

# Security in Context-aware Mobile Business Applications

Inauguraldissertation  
zur Erlangung des akademischen Grades  
eines Doktors der Naturwissenschaften  
der Universität Mannheim

vorgelegt von  
M.Sc. Emin Islam Tath  
aus Istanbul (Türkei)

Mannheim, 2008

Dekan: Professor Dr. Felix Freiling, Universität Mannheim  
Referent: Professor Dr. Stefan Lucks, Bauhaus-Universität Weimar  
Korreferent: Professor Dr. Matthias Krause, Universität Mannheim

Tag der mündlichen Prüfung: 06. März 2009

# Abstract

The support of location computation on mobile devices (e.g. mobile phones, PDAs) has enabled the development of context-aware and especially location-aware applications (e.g. Restaurant Finder, Friend Finder) which are becoming the new trend for future software applications. However, fears regarding security and privacy are the biggest barriers against their success. Especially, mobile users are afraid of the possible threats against their private identity and personal data.

Within the M-Business research group at the University of Mannheim, various security and privacy aspects of context-aware mobile business applications are examined in this thesis. After providing a detailed introduction to context-aware applications, the security challenges of context-aware applications from the perspectives of different principals (i.e. mobile users, the broker, service providers) are analyzed. The privacy aspects, the challenges, the threats and legal directives regarding user privacy are explained and illustrated by real-life examples. The user-centric security architectures integrated within context-aware applications are introduced as anonymity and mobile identity management solutions. The M-Business security architecture providing security components for communication security, dynamic policy-based anonymity, secure storage on mobile devices, identity management for mobile users and cryptography libraries is explained in detail. The LaCoDa compiler which automatically generates final Java code from high level specifications of security protocols is introduced as a software-centric solution for preventing developer-specific security bugs in applications.



# Zusammenfassung

Die Funktionalität der Positionsbestimmung auf mobilen Geräten (z.B. PDAs, Handys) unterstützt die Entwicklung kontext- und besonders ortsbezogener Anwendungen (z.B. Restaurant-Finder, Freund-Finder), die sich als neuer Trend auf dem Softwaremarkt abzeichnen. Eine wesentliche Barriere für den kommerziellen Erfolg dieser Anwendungen sind jedoch die Sorgen der Benutzer um ihre Sicherheit und Privatsphäre. Besonders mobile Benutzer fürchten die Preisgabe ihrer Identität und ihrer persönlichen Daten.

Die vorliegende Arbeit ist im Umfeld der Mobile Business Forschungsgruppe an der Universität Mannheim entstanden und betrachtet verschiedene Sicherheitsaspekte kontextbezogener Mobile Business Anwendungen. Im Anschluss an eine detaillierte Definition und Beschreibung kontextbezogener Anwendungen werden deren Sicherheitsrisiken aus der Perspektive von mobilen Benutzern, Brokern und Diensteanbietern untersucht. Die verschiedenen Aspekte der Privatsphäre, zugehörige Risiken und Bedrohungen sowie gesetzliche Vorschriften werden erläutert und mit Beispielen aus der Praxis illustriert.

Im nächsten Schritt werden benutzerbezogene Sicherheitsarchitekturen vorgestellt, die in kontextbezogene Anwendungen als Lösungen für Anonymität und mobiles Identitätsmanagement integriert werden können. Darauf aufbauend wird eine Mobile Business Sicherheitsarchitektur beschrieben, die verschiedene Sicherheitskomponenten für Kommunikationssicherheit, dynamische policybasierte Anonymität, sichere Datenhaltung auf mobilen Geräten, mobiles Identitätsmanagement und kryptographische Bibliotheken bereitstellt. Schließlich wird der kryptographische Compiler LaCoDa diskutiert, der Spezifikationen kryptographischer Protokolle automatisch in Java-Code übersetzt und damit das Risiko implementationsbedingter Sicherheitslücken verringert.



# Acknowledgments

It is a great pleasure and honor to write the acknowledgment page of the thesis. It is not possible to forget the people who have contributed to this thesis and have always supported me during the difficulties of the PhD time.

First and foremost, my special thanks go to my two PhD supervisors, Prof. Dr. Stefan Lucks and Prof. Dr. Matthias Krause. With his valuable knowledge and background in security and especially cryptography/cryptanalysis, Stefan has become a superb guide for me, enabling me to comprehend security from the perspectives of attackers. I have realized the real meaning of cryptography during discussions with him and also from the supervision of the exercise sessions of the cryptography lectures. Beyond his technical support, he has always encouraged and helped me to attend conferences, workshops and summer schools which have enhanced my technical skills and also extended my social network in the security community. I can never forget that he has struggled to find the financial support for me to complete this thesis<sup>1</sup>. I want to say finally that the German term “Doktorvater” which can be translated as “PhD father (i.e. PhD supervisor)” is a great fitting description which summarizes Stefan’s value for me. Handling all my administrative problems, Matthias has encouraged me to focus on my research and to complete this PhD. I also appreciate his close and sincere relations with me and other colleagues in the chair.

Additional thanks go to my former colleague and good friend Dirk Stegmann. Discussing research topics and working in the M-Business research group together with him has facilitated productive results. Without his advice and support, my PhD time would have been much harder.

I want to thank also all the professors and colleagues in the M-Business research group in Mannheim. It was a great experience to become a part of their research activities and to take responsibility for security research.

---

<sup>1</sup>This work was partially supported by the Landesstiftung Baden-Württemberg and the Ministry of Science, Research and Arts of the State of Baden-Württemberg.

My special thanks go to my family, especially to my parents. Over many years they have lovingly brought me up and have provided me with immense support, both before, and during the time, I have been writing my PhD thesis.

I wish to dedicate this thesis to my wonderful wife Arzu. Her patience and perseverance have helped me to focus on the thesis and bring it to a successful conclusion. I very much appreciate everything (and that is very much) that she has contributed to my life.



To my wife Arzu



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Mannheim Mobile Business Research Group . . . . .	3
1.2	Security and Privacy . . . . .	6
1.3	Thesis Structure and Contributions . . . . .	8
1.3.1	Content . . . . .	8
1.3.2	Publications . . . . .	9
<b>2</b>	<b>Context-aware Applications</b>	<b>11</b>
2.1	Common Terms . . . . .	11
2.1.1	Context . . . . .	11
2.1.2	Context-awareness . . . . .	13
2.1.3	Location-awareness . . . . .	13
2.1.4	E-Business, M-Business and M-Commerce . . . . .	13
2.2	Context-aware Mobile Business Applications . . . . .	14
2.2.1	Function Logic . . . . .	15
2.2.2	Pull vs. Push Services . . . . .	16
2.2.3	Categorization . . . . .	16
2.3	Existing Projects . . . . .	18
2.4	Target Application Scenarios . . . . .	19
2.4.1	Restaurant Finder . . . . .	20
2.4.2	Friend Finder . . . . .	22
2.5	Location Determination . . . . .	22
<b>3</b>	<b>Security Analysis</b>	<b>25</b>
3.1	Information Security Principles . . . . .	25
3.1.1	Confidentiality . . . . .	25
3.1.2	Integrity . . . . .	25
3.1.3	Availability . . . . .	26
3.2	Security Challenges . . . . .	26

3.2.1	Content and Communication Anonymity . . . . .	26
3.2.2	Privacy of Personal Data . . . . .	28
3.2.3	Location-based Spamming . . . . .	30
3.2.4	Integrity and Authenticity of Service Descriptions and Results . . . . .	31
3.2.5	Authentication and Authorization . . . . .	32
3.2.6	Confidentiality of the Communication . . . . .	33
3.2.7	Confidentiality of Locally Stored Data . . . . .	34
3.2.8	Secure Software Development . . . . .	35
3.2.9	Usability vs. Security . . . . .	36
3.2.10	Secure Mobile Payment and Fair Exchange . . . . .	37
3.2.11	Rogue Access Points and forged GPS-signals . . . . .	38
3.3	Extra Limitations . . . . .	39
<b>4</b>	<b>Exploits against User Privacy</b>	<b>41</b>
4.1	Privacy Challenges . . . . .	41
4.1.1	Location- and Action-relevant Risks . . . . .	42
4.1.2	Relationship-relevant Risks . . . . .	43
4.1.3	Monetary Risks . . . . .	43
4.1.4	Medical Data Risks . . . . .	44
4.1.5	Dynamic Pricing . . . . .	44
4.2	Legal Directives . . . . .	45
4.2.1	The Directive 95/46/EC (Data Protection) . . . . .	45
4.2.2	The Directive 2002/58/EC (E-Privacy) . . . . .	46
4.3	Privacy Threats in the Media . . . . .	47
4.4	A Case Study: Google Hacking . . . . .	50
4.4.1	Google Search Parameters . . . . .	52
4.4.2	Exploits against Personal Data . . . . .	53
	Identification Data . . . . .	53
	Sensitive Data . . . . .	55
	Confidential Data . . . . .	56
	Secret Data . . . . .	58
4.4.3	Attempts to obtain Cryptographic Secrets . . . . .	58
	Hashed Passwords . . . . .	59
	Secret Keys . . . . .	59
	Public Keys . . . . .	60
	Private Keys . . . . .	60
	Encrypted Files . . . . .	60
	Signed Messages . . . . .	61
4.4.4	Countermeasures . . . . .	62

4.4.5	TrackingDog: A Penetration Testing Tool for Privacy	63
4.4.6	Related Tools . . . . .	64
4.4.7	Discussion . . . . .	65
<b>5</b>	<b>User-centric Proposed Solutions</b>	<b>67</b>
5.1	The SALSA Client Security Architecture . . . . .	67
5.2	Dynamic Anonymity . . . . .	69
5.2.1	Existing Solutions for Anonymity . . . . .	71
5.2.2	New Anonymity Challenges . . . . .	74
	Limited Hardware Capabilities . . . . .	75
	Dynamic Anonymity . . . . .	76
5.2.3	Towards A Solution . . . . .	76
	Anonymity Parameters . . . . .	77
5.2.4	The Architecture . . . . .	78
	The Fat Client Architecture . . . . .	78
	The Thin Client Architecture . . . . .	79
	Policies and Templates . . . . .	81
5.2.5	Threat Analysis . . . . .	84
5.2.6	Strengths and Weaknesses of the Architectures . . . . .	85
5.2.7	Future Work . . . . .	85
5.3	Mobile Identity Management . . . . .	86
5.3.1	Related Work of Privacy and Identity Management . . . . .	87
5.3.2	User-controlled Mobile Identity Management . . . . .	89
5.3.3	Privacy Policy in P3P . . . . .	91
	Shortcomings of P3P/Appel . . . . .	92
	Extensions to P3P/Appel . . . . .	95
5.3.4	The Aspects . . . . .	96
	Context-to-Context Dependence . . . . .	97
	Context-to-Context Relation . . . . .	97
	Blurring in Levels . . . . .	98
	Extensible Preference Language . . . . .	99
	Trust Management with P3P . . . . .	100
	Status as Soft Shut-Down Button . . . . .	101
	History Management . . . . .	101
	Confidential Data Management . . . . .	101
	Content and Communication Anonymity . . . . .	102
5.3.5	Integration of the Aspects . . . . .	102

<b>6</b>	<b>Software-centric Proposed Solutions</b>	<b>105</b>
6.1	Software Engineering for Security . . . . .	105
6.2	LaCoDa: The Cryptographic Compiler . . . . .	106
6.2.1	The Architecture . . . . .	107
6.2.2	The Specification Language . . . . .	108
6.2.3	Template File . . . . .	109
6.2.4	Concrete Example: Encrypt-then-Authenticate Pro- tocol . . . . .	110
6.2.5	Implementing Security Protocols . . . . .	112
6.2.6	Discussion . . . . .	114
<b>7</b>	<b>Conclusion and Future Work</b>	<b>115</b>
<b>A</b>	<b>Acronyms</b>	<b>135</b>
<b>B</b>	<b>LaCoDa Sample Output Code</b>	<b>139</b>
B.1	Class File for EncryptThenAuthenticate_A.java . . . . .	139
B.2	Class File for EncryptThenAuthenticate_B.java . . . . .	140
<b>C</b>	<b>Extended Backus-Naur-Form</b>	<b>143</b>

# List of Figures

1.1	A Vision of Context-aware Mobile Applications . . . . .	2
1.2	Interaction of Security and Other Research Groups . . . . .	5
2.1	Context Feature Space . . . . .	12
2.2	The Function Logic of Context-aware Applications . . . . .	15
2.3	SALSA Demo Application - Search Page . . . . .	19
2.4	SALSA Demo Application - Context Settings . . . . .	20
2.5	SALSA Restaurant Finder Demo . . . . .	21
2.6	Friend Finder Application . . . . .	22
4.1	Fragments of Identity . . . . .	42
4.2	A private Chat Log . . . . .	49
4.3	Dumped Passwords . . . . .	50
4.4	Private Emails . . . . .	51
4.5	Confidential Documents . . . . .	52
4.6	The robots.txt file of www.whitehouse.gov . . . . .	58
4.7	TrackingDog - Main GUI . . . . .	63
4.8	TrackingDog - Result GUI . . . . .	64
4.9	SiteDigger - SiteDigger Google Hacking Scanner . . . . .	65
5.1	The SALSA Client Security Architecture . . . . .	68
5.2	The SALSA Client Security Architecture (extended) . . . . .	70
5.3	The SALSA Client Security Architecture - Secure Communi- cation . . . . .	71
5.4	Jap Architecture . . . . .	74
5.5	Tor Architecture . . . . .	74
5.6	The Fat Client Anonymity Architecture . . . . .	79
5.7	The Thin Client Anonymity Architecture . . . . .	80
5.8	Samples of Anonymity Templates . . . . .	82
5.9	Examples of Anonymity Policies . . . . .	83

5.10	P3P Sample Policy . . . . .	92
5.11	P3P Sample Policy (cont.) . . . . .	93
5.12	Privacy Concerns of Users . . . . .	94
5.13	Feature Relations for Privacy . . . . .	96
5.14	The Structure of Exceptions for Privacy Preferences . . . . .	99
5.15	A Sample of Privacy Preferences for Location . . . . .	100
5.16	A Sample of Privacy Preferences for Interests . . . . .	100
5.17	The Aspects integrated in the Friend Finder Application . . .	103
6.1	The Architecture of LaCoDa . . . . .	108
6.2	Template File Example . . . . .	110
6.3	The protocol specification of the Encrypt-then-Authenticate .	111



# List of Tables

2.1	Categorization of Context-aware Applications . . . . .	17
3.1	List of Security Challenges (U:User, B:Broker, SP:Service Provider, +:challenge, -:no chal- lenge) . . . . .	27
5.1	Performance of Cryptographic Operations . . . . .	75
C.1	The Reserved Words . . . . .	143
C.2	The symbols and operators used in the EBNF specification .	144



# Chapter 1

## Introduction

Home users became acquainted with the Internet from the middle of the 90's by surfing static web pages, checking e-mails and chatting with friends. In the Internet era, the technology has rapidly progressed. Today even small mobile devices such as mobile phones and personal digital assistants (PDAs) can even be used to communicate over the Internet. Additionally, wireless communication techniques (i.e. Wlan, Bluetooth, GPRS/UMTS, etc.) have been considerably improved. Mobility has introduced mobile applications. In spite of their limitations such as small screen size, low processor power and restricted battery life time, mobile devices running mobile applications have gained a large degree of acceptance among end users. Technology producers have also realized the huge business potential implicit within the mobile community. Today, context-aware applications that facilitate the context data (e.g. location, time, velocity, etc.) of mobile users for enabling services are the new trend for future mobile business applications. Service providers and telecommunication providers are already focusing on extending their infrastructures and developing applications in order to support mobile users so they may have the benefit of context-aware and especially location-aware applications.

Locating kids [34] and people in emergency [39], locating moving objects (e.g. fleet management) [20], location-based chat and games [40], indoor and outdoor routing [30], locating nearby restaurants, cinemas and gas stations are examples of already implemented context-aware and location-aware mobile business applications. A general view of context-aware mobile applications is depicted in Figure 1.1. In such applications, mobile user's context data such as current location, time, weather, profile, etc. are considered by service providers while the mobile users holding their PDAs are interested

in getting the service that best fits their current context.

Security has been always considered as being of major importance in the digital world. Security risks are present in all communications over the Internet. In the newspapers, radios, TVs and Internet media, many security breaches have been experienced causing business failures, financial loss and privacy violations (see Section 4.3). Even prison sentences as a punishment have become common in many situations [2, 57, 61]. For the continuing success of any business, data security is very important. Based on the questionnaires completed by a sample of mobile device users, it can be confirmed that security of personal data is the most critical issue for these users. As a conclusion, considering context-aware applications in which very sensitive location data is exchanged between the communicating parties, security and privacy of personal context data can be looked upon as being a major barrier against the potential successful implementation and acceptance of mobile business applications.



Figure 1.1: A Vision of Context-aware Mobile Applications<sup>1</sup>

Even though high-level mobile devices exist and wireless mobile communication methods have been improved, the development process of mobile business applications is quite young and is an open research issue. The Mobile Business Research Group [66] at the University of Mannheim has

<sup>1</sup>Referenced from <http://www.m-business.uni-mannheim.de/SALSA/Overview.htm>

focused on designing and developing a generic platform and server/client components for context-aware and especially location-aware mobile business applications since September 2004. As a participant in the research group, we have studied the security aspect in depth by analyzing the possible security and privacy threats for the main principals (i.e. the broker, mobile users and service providers) and designing and developing the required security protocols and components in terms of user-centric and software-centric solutions.

## 1.1 Mannheim Mobile Business Research Group

With the future trend towards mobile applications, the Mobile Business Research Group was formed at the University of Mannheim in Germany. The main focus of the research group is context-aware and especially location-aware mobile business applications. More concretely, we aim at designing a generic framework for context-aware applications. The framework provides the required infrastructure and server/client components in order to realize and deploy a specific context-aware application easily and rapidly.

Initially, there were 7 research chairs involved in the activities of the research group from the departments of computer science (Ger. Informatik), information systems (Ger. Wirtschaftsinformatik) and business administration (Ger. Betriebswirtschaftslehre). So far, the research group has completed the SALSA [56] and LAMBADA [35] projects successfully. The SALSA project (Software Architectures For Location-Specific Transactions in Mobile Commerce) aimed at creating an advanced infrastructure for developing and deploying location-based mobile commerce applications. It was supported by the State of Baden-Württemberg [36]. The LAMBADA project (Location-Aware Mobile Business Adhoc Architecture) focused on technology development for location-based and context-based mobile business applications and was supported by the Ministry of Science, Research and Arts of the state of Baden-Württemberg.

The research chairs and their contributions to the M-Business group can be summarized as follows:

- Chair of Software Technology (Prof. Dr. Colin Atkinson) [156]: Service Discovery is the focus of this research group. The problem of discovering location-based services dynamically has been studied. Developing location-based services within the context of service-oriented architectures, the development of algorithms for finding and evaluat-

ing services and matching them to user requirements are all activities which have been implemented by the group.

- Chair of Multimedia and Network Technology (Prof. Dr. Wolfgang Effelsberg) [161]: This research group deals with position determination for mobile devices. The existing localization technologies, especially the technologies for outdoor localization (i.e. GPS) and indoor localization (i.e. WLAN) have been analyzed. The research group also developed other localization algorithms for PDAs using GPS, WLAN and Bluetooth technologies.
- Chair of Database Technology (Prof. Dr. Guido Moerkotte) [158]: Developing ontology for describing location information is the main focus of this research group. The database has been developed by using the location ontology for places as a schema.
- Chair of Theoretical Computer Science (Prof. Dr. Matthias Krause, Prof. Dr. Stefan Lucks<sup>2</sup>) [160]: The research group has been involved in analyzing the security challenges in the M-Business framework, developing the required security components and providing the required cryptographic libraries for confidential communication and secure storage.
- Chair of Information System III (Prof. Dr. Dr. Martin Schader, Dr. Markus Aleksy) [157]: The research group has analyzed the existing technologies and software engineering methods for dynamic re-configuration of applications. Other contributions were finding new approaches and technologies that support the flexibility of mobile applications.
- Chair of Business Administration and Information Systems (Prof. Dr. Armin Heinzl) [155]: The goal of this research group is the selection of a generic application scenario for context-based applications using mobile technologies. The emphasis is on examining how the service-oriented, location-based and semantic aspects of location-based services can be incorporated in the conceptual design.
- Chair of Business Administration and Marketing II (Prof. Dr. Hans H. Bauer) [159]: Analyzing of location-based applications from the user requirements perspective has been the main concern of this research

---

<sup>2</sup>Prof. Lucks has meanwhile moved to the Bauhaus University of Weimar.

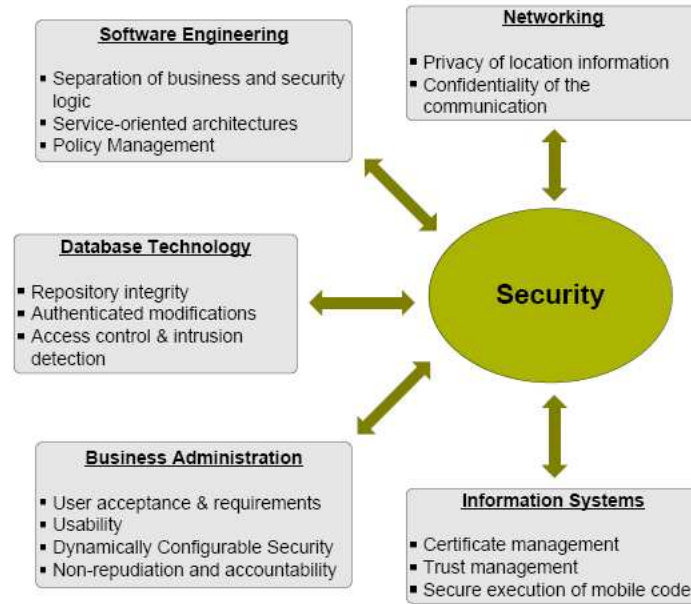


Figure 1.2: Interaction of Security and Other Research Groups

group. Especially, the existing literature with a business focus has been analyzed in detail and a catalog of user requirements has been prepared.

Interactions between the research chairs have been necessary within the M-Business projects. As the security group, we have assisted different groups in providing support for the security aspects of their projects as shown in Figure 1.2. Privacy and confidentiality of location information is required from the networking group. Separation of business and security aspects and policy-based service-oriented architectures are important requirements to be met by the software group. Integrity of databases, authentication and authorization of database modifications are required by the database group. Usability is a requirement from the business administration group and trust management is a requirement from the information systems group.

After completing the projects SALSA and LAMBADA successfully, the research group has started working on the GEM project [22] (Generic Environment for Mobile Business) which is supported by “Deutsche Forschungsgemeinschaft (DFG)” [14]. The project analyzes potential benefits and factors of adoption and acceptance as well as the development of technological

components for a generic reference architecture to support several kinds of key mobile business applications.

## 1.2 Security and Privacy

Security aims to protect information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction<sup>3</sup>. Privacy enables an individual or group to safeguard information about themselves and prevent their unintended release to others or to attackers.

Security and privacy are the non-functional requirements of a system, but they have a very critical role for the acceptance of the system. Considering context-aware applications in which a large quantity of personal user data is exchanged between the principals, security and privacy become inevitable and paramount. It can even be the case that a business can fail if the required countermeasures are not in place.

The M-Business research group has aimed at constructing a generic framework that fulfills the requirements of different context-aware applications. Security should be also a part of this framework. The security analysis should be done in order to specify the possible threats against the different principals (i.e. the broker, mobile users and service providers). Based on the results of the analysis, the required security protocols should be designed and the required security components should be developed and integrated within the M-Business generic framework.

Within the scope of security management tasks, the following security aspects have been studied:

- **Security Analysis:** In the analysis phase, the security threats in the M-Business framework and the possible solution mechanisms against these threats were specified. The results were published within a scientific paper [189] and presented in the M-Business workshop [65] in the poster session.
- **Security Questionnaire:** The Business Administration and Marketing II group [159] prepared a questionnaire for mobile users in order to get to know their opinions on different topics such as usability, usefulness and also security to enhance applications in the M-Business framework. We contributed to this questionnaire by integrating our security analysis results within the questionnaire.

---

<sup>3</sup>Referenced from U.S. Code Collection Definitions -  
[http://www.law.cornell.edu/uscode/html/uscode44/usc\\_sec\\_44\\_00003542- - -000-.html](http://www.law.cornell.edu/uscode/html/uscode44/usc_sec_44_00003542- - -000-.html)



- **Privacy Exploits:** People may not realize the importance of their privacy unless their privacy is threatened or violated. If someone endangers their privacy, they would react vigorously. We have used the technique “Google Hacking” [24] for revealing sensitive cryptographic secrets (e.g. username-passwords, secret keys, private keys, encrypted messages, etc.) and personal confidential documents (e.g. confidential emails, forum postings, etc.). A tool namely *TrackingDog* [137] was also developed for making the Google searches automatically. This work has shown that users involved with context-aware applications should also be equipped with the relevant PETs (privacy enhancing tools) in order to protect themselves and guarantee their privacy.
- **User-centric Solutions:** Anonymity, location privacy, mobile identity management and trust management were the main security concerns for the user-centric solutions. We have designed an anonymity architecture [188, 190] and developed the anonymity components for enabling mobile user to communicate anonymously with the broker and service providers. We have designed a mobile identity management system that supports location privacy by blurring in levels and trust management by utilizing the P3P protocol [51]. P3P originally targets only Internet browsing, but not very suitable for integration directly within context-aware applications. The shortcomings of P3P for context-aware applications were analyzed and published in a scientific paper [185]. For software development, the M-Business research group use Java technologies. The Java security APIs do not run, however, very efficiently on mobile devices. We therefore ported cryptlib security library [10] on ARM-processors and this enabled very fast execution of security algorithms.
- **Software-centric Solutions:** Application developers cause many security bugs in implementations. They are not security experts and can easily make errors and endanger the security of their implementations. Buffer overflows, SQL injections and XSS injections are typical examples of such implementation errors. To eliminate these implementation bugs, we have aimed to automate this process. The LaCoDa compiler [181] we developed provides a specification language. The developer just encodes the relevant security or cryptography protocol for the LaCoDa in its specification language and it then creates the final Java source code. In the near future, it will support more languages - such as C, Ada and even the formal verification languages.

## 1.3 Thesis Structure and Contributions

### 1.3.1 Content

This thesis emphasizes the fundamentals of context-aware and location-aware applications, security and privacy analysis of applications within the M-Business framework and the security architecture (i.e. security components, libraries) integrated within the framework. User-centric proposed solutions (i.e. anonymity, privacy, identity management) and software-centric proposed solutions (i.e. automatics code generation from security protocols) are explained and illustrated with concrete examples in the thesis in detail.

The thesis is organized as follows:

- Chapter 2 explains the definitions of some common terms like context, context-awareness and location-awareness. The differences between electronic business, mobile business and mobile commerce are explained. Existing context-aware and location-aware applications are introduced. The main principals and the different types and categorization of context-aware applications are also mentioned in this chapter. Two target demo applications (i.e. Restaurant Finder and Friend Finder) are introduced. Finally, the chapter concludes with an explanation of the location determination techniques for outdoor and indoor applications.
- Chapter 3 explains the main objectives (i.e. confidentiality, integrity and availability) in terms of security and the security challenges in context-aware applications from the perspectives of the broker, mobile users and service providers. Some additional security limitations are given in the conclusion to this chapter.
- Chapter 4 focuses on privacy aspects. Privacy challenges of location-aware applications and their legacy aspects are described in detail. Real life privacy threats as detailed in the media are given as well. Finally, real life privacy threats are illustrated by using Google hacking techniques which are used to reveal confidential and secret personal information containing even private keys or private emails. The countermeasures for Google hacking, the relevant tools and our penetration testing tool, namely TrackingDog, are introduced at the end of this chapter.
- Chapter 5 focuses on our user-centric solutions for M-Business security architecture, anonymity and mobile identity management. The

SALSA security architecture, which provides mechanisms and components for secure communication, secure storage, anonymous communication and mobile identity management, is explained in detail. Enhancements to the existing Mix-net based anonymity networks, new aspects for mobile identity management from the context-aware application perspective and the extensions to P3P privacy policies and preferences are explained in this chapter.

- Chapter 6 mentions the common mistakes in software engineering from the security perspective. It introduces our cryptography compiler La-CoDa [181], which improves the software development process by automating Java code generation from high-level specifications of security protocols. The architecture of the compiler, the syntax and features of the specification language and concrete code generation examples are explained in this chapter.
- Chapter 7 concludes the thesis.

### 1.3.2 Publications

The contents of this thesis are based on a number of publications by the author, as follows:

#### Peer-Reviewed Papers:

1. E.I.Tatlh and S. Lucks: Mobile Identity Management Revisited. In Proceedings of the 4th International Workshop on Security and Trust Management, Trondheim-Norway, June 2008.
2. E.I.Tatlh: Privacy in Danger: Let's google your privacy. In Proceedings of the Third IFIP WG 9.2, 9.6/11.6, Series: IFIP International Federation for Information Processing , Vol. 262, Simone Fischer-Hübner, Penny Duquenoy, Albin Zuccato, Leonardo Martucci (Eds.), Boston: Springer, pp. 51-59, June 2008.
3. E. I. Tatlh: Google Hacking for Privacy, Third International Summer School The Future of Identity in the Information Society, Karlstad-Sweden, 6-10 August 2007.
4. E. I. Tatlh: Extending P3P/Appel for Friend Finder, The International Workshop on Privacy-Aware Location-based Mobile Services (PALMS), Mannheim, 11 May 2007.

5. E. I. Tatl: Privacy in Context-aware Mobile Business Applications, IADIS International Conference (E-Commerce '06), Barcelona-Spain, 9-11 December 2006.
6. E. I. Tatl: Context Data Model for Privacy, PRIME Standardization Workshop, IBM Zürich Research Center, 6-7 July 2006.
7. E.I. Tatl, D. Stegemann and S. Lucks: Dynamic Anonymity. The 4th World Enformatika Conferences, International Conference on Information Security (ICIS'05), Istanbul-Turkey, 2005.
8. S. Lucks, N. Schmoigl and E.I.Tatl: Issues on Designing a Cryptographic Compiler. WEWoRC (Western European Workshop on Research in Cryptology), Leuven-Belgium, 2005.
9. E. I. Tatl, D. Stegemann and S. Lucks: Security Challenges of Location-Aware Mobile Business. The second IEEE International Workshop on Mobile Commerce and Services (WMCS'05), München-Germany, 19 July 2005.

**Non-Reviewed Papers:**

10. M. Kessler, S. Lucks, E. I. Tatl: TrackingDog - A Privacy Tool against Google Hacking, 7. Kryptotag, Bonn, 9. November 2007.
11. E. I. Tatl: Google Reveals Cryptographic Secrets, Technical Report of 1. Kryptowochende, Kloster Bronbach, 01-02 July 2006.
12. E. I. Tatl, D. Stegemann and S. Lucks: Dynamic Mobile Anonymity with Mixing, Technical Report, University of Mannheim, March 2006.

## Chapter 2

# Context-aware Applications

### 2.1 Common Terms

The common terms (i.e. context, context-awareness, location-awareness, location-based, location-specific, mobile business, mobile commerce and electronic business) that are used through the thesis are explained in this section. Additionally, the following questions are answered in order to understand these terms and how they relate to each other:

- What is meant by the term *context*?
- What is context-awareness and location-awareness?
- What are the relations between the terms location-aware, location-based and location-specific?
- What are the differences between context-aware and location-aware applications?
- What is the difference between electronic business, mobile business, electronic commerce and mobile commerce?

#### 2.1.1 Context

No common definition of the term *context* exists in the literature. Abowd et al. describe context as “*any information that can be used to characterize the situation of an entity*” [85] and an entity is “*a person, place or object that is considered relevant to the interaction between the user and an application, including the user and the application themselves*”. Beresford defines four

main context of an entity: identity, location, activity and time [97]. According to Schilit [168], a user's context includes information such as health, mood, schedule, level of mobility (e.g. scooter, bike and car) and location.

In context-aware applications, location information has such a dominating position over other context data that the terms context-aware and location-aware are mostly used interchangeably. Considering this misinterpretation, Schmidt et al. propose another model for the representation of context data in their paper namely "*There is more to context than location*" [169]. In their model as depicted in Figure 2.1, context can originate from *human factors* or the *physical environment*. Human factors can be categorized into three as *user* (his/her knowledge, characteristics, habits, etc.), *social environment* (social interaction, etc.) and *tasks* (tasks engaged in, general goals, etc.). The physical environment can be divided into three categories as *conditions* (light, auditory stimulus, temperature, etc.), *infrastructure* (surroundings for computation and communication) and *location* (absolute location, relative location, etc.).

The M-Business framework which supports different context-aware applications requires the managing of all context data involved in the applications. Static data (e.g. forename, surname, address, etc.), dynamic data (velocity, light level, air pressure, etc.), entity relations, location, time and user morale are all instances of context data and need to be supported by the M-Business framework. Schmidt et al.'s data model meets our requirements for the M-Business framework.

In contrast this model does not take any privacy aspect into consideration. It has been extended and enhanced towards a more privacy aware model [183]. This topic is explained in detail in Section 5.3.3.

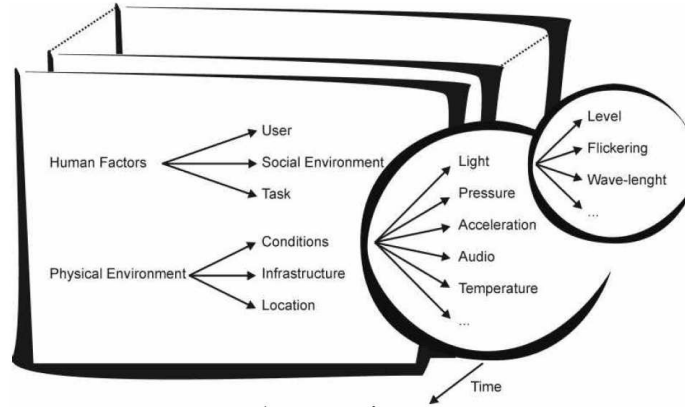


Figure 2.1: Context Feature Space<sup>1</sup>

### 2.1.2 Context-awareness

Dey defines a system as context-aware if “*it uses context to provide relevant information and/or services to the user, where relevancy depends on the users task*” [104]. That means context-aware applications use context as input when delivering a service. This input can be either given explicitly by users or explicitly retrieved and forwarded to service providers.

As a concrete example, a service which provides a list of particular restaurants for a given location and/or given preferences (e.g. non-smoking place) is a typical *context-aware application*. The location information can either be entered by the user or automatically computed (i.e. based on GPS, WLAN, or Bluetooth technologies). Context-aware *mobile* applications are more specialized types of context-aware applications. The users of context-aware mobile applications utilize mobile devices (e.g. mobile phones and PDAs) in order to receive services which are especially enhanced for mobile devices and mobile users.

### 2.1.3 Location-awareness

Location-aware applications are a subset of context-aware applications, but have their focus on location information. They may still retrieve other context data, but their concentration on location makes them being called as location-aware applications. The term *location-aware* is used even more frequently than context-aware in the literature. This is to emphasize the introduction of new functionalities dependent on the *automatic retrieval* of location information. In addition to the term location-aware, the terms location-based and location-sensitive are also used interchangeably in the literature.

Even though the M-Business framework considers a very broad scope of context data, the location data and therefore location-aware applications have prime importance for our research and development project.

### 2.1.4 E-Business, M-Business and M-Commerce

Even though Electronic Business (E-Business), Mobile Business (M-Business) and Mobile Commerce (M-Commerce) have certain similarities and are commonly used interchangeably in the literature, they all cover different application areas.

---

<sup>1</sup>Referenced from [169]

E-Business is defined as “*broadly as any business process that relies on an automated information system. Today, this is mostly done with Web-based technologies*” (from Wikipedia).

Möhlenbruch and Schmieder compare M-Business and M-Commerce in [144]: “As a new characteristic of E-business, M-Business enables development of business processes through financial use of wireless transport technologies over mobile devices and thus affects many E-Business domains such as E-Commerce, E-Procurement, Supply Chain and Customer Relation Management. On the other hand, M-Commerce is involved only with monetary transactions that are executed by wireless information transfer using mobile devices”. Gerpott defines M-Business as “a generic term of M-Commerce, but which covers additional support of non-financial exchange processes in and between companies via mobile information services” [191]. Link and Schmidt define M-Commerce as “electronic supported development of business communication and transaction processes via mobile end devices” [123]. Nicolai and Petersmann comment on M-Business and M-Commerce in [147]: “M-Business is understood as a subset of E-Business. When E-Business activities are carried out via a mobile device over a mobile network, then this is called M-Business. If the E-Commerce transactions are carried out via mobile devices, this is called M-Commerce”. MacDonald [136] summarizes M-Commerce as “making money through the phone”. Based on these explanations, it can be concluded that M-Business is a subset of E-Business, but its electronic transactions are only executed via mobile devices. M-Commerce is also a subset of M-Business, but its electronic transactions are only relevant to monetary transactions.

In conclusion, M-Business can be defined as business transactions over mobile telecommunication networks that are executed via mobile devices such as PDAs, mobile telephones, wireless-enabled laptops etc. By M-Commerce, we denote the subset of M-Business that involves commercial transactions, i.e. the exchange of material goods [167]. Selling books and CD’s over the Internet is a well-known example of M-Commerce, whereas a service for locating a person having a heart attack and sending an ambulance to him/her would be considered as M-Business rather than M-Commerce.

## 2.2 Context-aware Mobile Business Applications

More advanced mobile devices will become part of our daily lives in the near future. Many more context-aware and location-aware services will be implemented, and many more mobile transactions will be executed over mobile



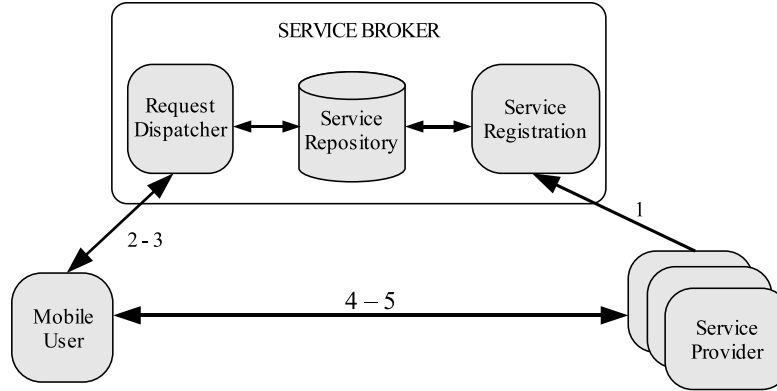


Figure 2.2: The Function Logic of Context-aware Applications

networks. Context-aware applications can be deployed for very different application logics (e.g. restaurant finder vs. fleet management). It is the case that their principals are the same and their function logics are very similar.

### 2.2.1 Function Logic

In context-aware applications, three main principals exist: *the broker*, *mobile users* and *service providers*.

Mobile users with mobile devices are interested in receiving M-Business services. A service provider offers chargeable and/or free services for mobile users and registers the service interfaces with the broker. The broker maintains a repository of available services and provides mobile users with the descriptions of services that are most suitable for the user according to user's preferences and context. For certain applications, it is also possible that the broker takes in addition the role of a service provider and directly offers services to mobile users. For data transmission, the framework utilizes the services of infrastructure providers such as telecommunication utilities.

The function logic illustrated in Figure 2.2 works as follows:

1. Service providers apply to the broker in order to register their services within the repository of the broker. In the case that the broker approves this application, the service becomes available to mobile users.
2. Mobile users choose a service category (e.g. restaurants) through the interface of the application(s) running on their mobile devices and query the broker for available services in this particular category. If

the user is interested in location-based services, he/she sends his/her location information to the broker along with the request. The user also sends his/her profile, which represents personal preferences and special interests.

3. Based on the profile, the broker determines the relevant services within its repository and sends back the corresponding service descriptions. A service description typically includes information about price, provider location and quality descriptions.
4. Upon getting the service descriptions, the user decides on a service and applies to the provider of this particular service.
5. The service provider charges the user (if appropriate), and provides the service.

### 2.2.2 Pull vs. Push Services

Context-aware applications can be grouped as *pull* and *push* services according as to how the service is conveyed to mobile users [194]. In pull services, users find the relevant service provider with the help of the broker and then request the service provider to obtain the service. In push services, users find the relevant service provider via the broker as in pull services, but afterwards request to “*join*” to a service and send regularly specific context data such as location to the provider. The service providers use this context data to deliver their services to the mobile users in the case of a certain event occurring (related to the context data) or a certain time period has elapsed.

### 2.2.3 Categorization

Based on the research results and existing literature analysis of Bauer et al. [119], there are 6 types of context-aware applications as depicted in Table 2.1.

Tracking services cover exact localization of persons or objects. Child tracking [34], localization of car drivers having accidents [37], blind guidance systems [7], tracking of seniors/persons in need [70], localization of friends, relatives, family members within the same geographical area [38], location-based chats [40] and group management via tracking of group members [27] are existing examples of person tracking services. Auto search [5], mobile phone searching [44], automatic localization of taxis [94] and fleet management [20] are the examples of existing object tracking services.

Main Category	Sub-categories
1- Tracking services	a) Person tracking services b) Object tracking services
2- Navigation services	a) General navigation services b) Special navigation services
3- Information services	a) General information services b) Interactive information services
4- Communication services	a) B2C communication services b) B2B communication services
5- Entertainment services	
6- Transaction services	

Table 2.1: Categorization of Context-aware Applications

Navigation services direct mobile users from their current location to a target location. If the user specifies the target location specifically, then it is a type of *general* navigation service. If the user is interested in being navigated to a particular service destination (e.g. restaurants, gas stations, etc.), then this is called a special navigation service. Adjustment of a traffic routing plan [121] and indoor routing in fairs [31] are examples of general navigation services. Navigation to near-by restaurants, shopping centers, gas stations [45, 46] and ATM locators [4] are examples of special navigation services.

Information services provide information relevant to the user's current location. With interactive information services, users can also react interactively after receiving the information. As an example, users can order pizza directly when they receive the details of the nearest pizza shops. Mobile city guides [43], weather information [75], traffic information [71] and time-tables of public transport with location plans and real-time information (e.g. delays) [92] are examples of general information services. Event information systems with location plans and friend-invite features [91] are an example of an interactive information service.

Communication Services as B2C (business-to-client) and B2B (business-to-business) aim at optimizing the communication possibilities between users and business enterprises [107, 166].

Entertainment services include location-based games [41], location-based radio, location-based storytelling, etc.

Transaction services occur when financial transactions are executed be-

tween the principals. Automated ticketing [121], buying products from billboards, and local advertisements are examples of transaction services.

## 2.3 Existing Projects

There are several existing research projects on context-aware and location-aware applications. Their focus is not specifically M-Business oriented.

The aim of the Nexus project [47] is modeling the spatial world for a variety of commercial and non-commercial context-aware applications by collecting context information from a number of sensors placed around and on stationary objects (e.g. streets, buildings) or mobile objects (e.g. people, vehicles). Location information relating to the relevant objects is stored by the central location service provider, and other objects or service providers can search the location of particular objects. Securitywise, Nexus focuses mostly on privacy issues. To control unauthorized access to location information, a certificate-based access control solution based on SPKI (simple public key infrastructure)[58] has been developed within Nexus. For querying the location of an object, the requester must hold an authorization certificate that is issued and digitally signed by this particular object. For personal data management Nexus uses virtual IDs. The identity manager from [122] is suggested for handling virtual IDs.

The WASP project [74] aims at developing a context-aware service platform based on web services. The aim of the project is to make it easier for service providers to manage their services and also easier for mobile users to search for, find and receive the relevant services. WASP proposes a P3P-based (*Platform for Privacy Preferences*) [51] architecture to guarantee the privacy of mobile clients. The P3P protocol enables web users to be aware of what kind of information is collected when they communicate with service providers. P3P-enabled user agents get the requests from users and ask the service providers for their P3P policy, and more specifically, what kind information they require from the user. Upon receiving the P3P policy, the agents compare the policy and the user privacy preferences. If there is no conflict, the agents send the relevant data to the providers, get the responses and forward them to the users.

The Nimbus project [48] provides a framework to support developers of location-based services. It presents a common interface for location data and hides the details of how location is captured. Switching between outdoor positioning systems (e.g. GPS) and indoor positioning systems (e.g. based on Bluetooth or WLAN) can be realized without affecting the context-aware

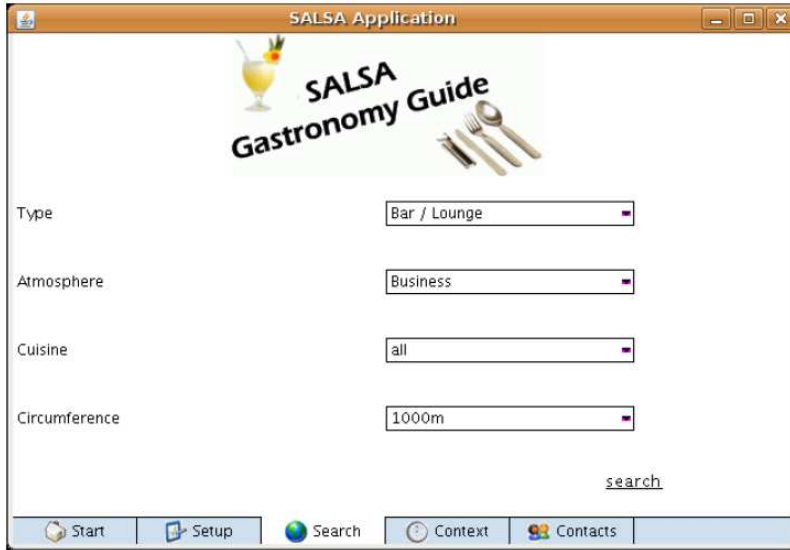


Figure 2.3: SALSA Demo Application - Search Page

applications. From the security point of view, the focus of the Nimbus is especially on network and communication security [93].

The OpenLS project [50] of the Open GIS consortium [68] provides common interfaces for the developers and providers of location-aware applications such as map and navigation applications.

The Nidaros framework [67] provides components for fast development of location-aware applications, i.e. especially applications like location-dependent advertisements, city guides and guides to tourist attractions.

## 2.4 Target Application Scenarios

Within the scope of SALSA project [56], the M-Business research group has implemented a demo application for finding restaurants which are near to mobile user's current location in Mannheim and which fits in with the user's preferences. The aim of this demo was to show that the architectures, methods and components (designed and developed within a generic context-aware framework by the different research chairs) can be used to build up a real-life application.

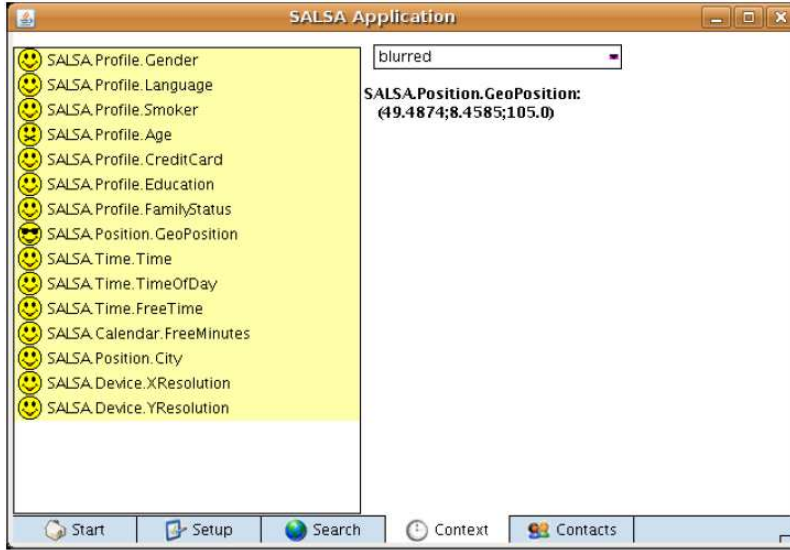


Figure 2.4: SALSA Demo Application - Context Settings

### 2.4.1 Restaurant Finder

*Restaurant Finder* is a very typical context-aware application providing gastronomy guide. It helps mobile users to locate the most suitable restaurants in the vicinity, based on user profiles and context data.

The SALSA demo application shown in Figure 2.3 provides Restaurant Finder as a service. In the setup menu, mobile users can set main profile and application configurations such as age, gender, GUI language, family status etc. In the context menu shown in Figure 2.4, different context data (e.g. location, education, credit card, free time, etc.) are displayed and configured.

Figure 2.5 depicts how the interactions between the broker, mobile users and restaurant providers are carried out. The broker is named Universal SDS (service discovery service) within the Salsa architecture. A mobile user who wishes to have a delicious lunch in a restaurant needs to use the search menu. The profile containing user preferences such as non-smoker, indoor/outdoor and restaurants within one kilometer radius are forwarded to the universal SDS along with the user's location. The universal SDS gets this service request and searches for the relevant restaurant finders within its repository. The Gastro Guide as a service provider has already



Thanks to the extensible SALSA architecture, other applications are also supported by the Salsa demo application. Event guide application provides mobile users with the ability to search for concerts, exhibitions, sport events, etc. Not only location information but also the available free time of users, weather information and profile data matching different events are also taken into consideration during the search process. The Tourist Guide application provides context-based sightseeing tours. Bargain Hunter application can offer users bargains and bonus coupons based on their context and location data.

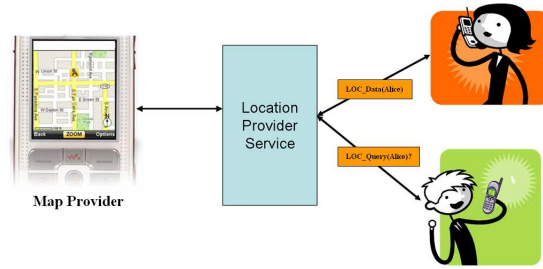


Figure 2.6: Friend Finder Application

### 2.4.2 Friend Finder

Even though the SALSA project chose the Restaurant Finder application (which is a type of *pull* service) as the demo application to develop, more complicated applications were needed to illustrate the existing privacy and security risks. Therefore, *Friend Finder* (which is a type of *push* service) has been also within our focus. Unlike the Restaurant Finder, interactions among mobile users are supported in the Friend Finder application and this causes more privacy risks that need to be taken into consideration.

Friend Finder is an example of a “person locating” service and is a location-aware application [28]. As illustrated in Figure 2.6, mobile users participating in the Friend Finder service regularly send their location data computed by their mobile devices to the central location provider and can also query the location of their particular friend through the location provider.

## 2.5 Location Determination

Location information is the most important context data for context-aware applications. It can be a concrete value (i.e. coordinates in 3D dimension) or a relative position (e.g. room number or building number). Mobile devices can discover their current geographical positions with the help of a special hardware device. Computing the location can be categorized into outdoor and indoor computing according to the underlying communication infrastructure.

Outdoor computing relies mostly on the availability of satellite-based positioning systems such as GPS (Global Positioning System) [124]. GPS uses a constellation of between 24 and 32 geostationary satellites, which transmit precise microwave signals. A GPS receiver receives these signals



from different satellites, and calculates the current position coordinates by timing the signals. In addition to location, receivers can determine their speed, direction, and the current time. The EU-project Galileo [21] is an alternative to GPS and is expected to be operational by 2013.

GPS is not very effective for indoor areas, since the communication between GPS receivers and satellites is very often broken and therefore location is not always able to be determined. For indoor areas, the existing wireless network infrastructures (i.e. WLAN, Bluetooth) often enable a cheaper and more accurate computation of the location [153]. These technologies can help mobile devices to determine their location relative to the location of access points in indoor areas. There are at the present time a number of positioning systems for indoor areas such as Active Badge [196], Cricket [154] and Easy Living [101].

For more information about positioning systems, please refer to [128].



## Chapter 3

# Security Analysis

### 3.1 Information Security Principles

The CIA triad consisting of *confidentiality*, *integrity* and *availability* are the core principles of information security [180]. These principles are the main requirements for *secure* systems and *secure* information management.

#### 3.1.1 Confidentiality

Confidentiality provides the protection of sensitive information from *unauthorized access* [99]. Similarly, the ISO (International Organization for Standardization) defines confidentiality as “ensuring that information is accessible only to those authorized to have access”.

As an example, mobile users of the restaurant finder application should be able to send their credit card information (*as sensitive information*) to their banks in a *secure* way, guarded against *unauthorized disclosure* for the debit. Otherwise, an attacker eavesdropping the communication channel can ascertain both the secret and private credit card data relating to the mobile user. Similarly, service providers storing personal information such as name, address, phone number of users within their databases must guarantee the confidentiality of this information.

Confidentiality is enabled in practice thanks to modern cryptography, i.e. symmetric and asymmetric encryption techniques [173].

#### 3.1.2 Integrity

Integrity guarantees the prevention of *unauthorized modification* of data [174]. Both data that is kept on volatile or non-volatile storage and data

that is in transit requires to maintain its integrity.

As an example, the payment details (i.e. payment amount, credit card information) of mobile users in the Restaurant Finder application should not be altered during the transactions between mobile users, service providers and the banks. Additionally, as in the Friend Finder application, location data should be protected against unauthorized modification.

### 3.1.3 Availability

Availability assures that information resources are all accessible to legitimate users when required. DoS (Denial of Service) attacks [143] are the common techniques to cause an information resource to become unavailable.

For example, with the Friend Finder application, mobile users send their location data at regular intervals the location provider. Therefore, the location provider is expected to be always accessible. In the Restaurant Finder application, the bank's resources are also expected to be always online in order to prevent inconsistencies during payments.

## 3.2 Security Challenges

Considering context-aware applications, each principal, i.e. mobile users, the broker and service providers or other third parties can be potential attackers and threaten the security and privacy of the others. Therefore, the security threats should be considered and evaluated from the perspectives of all principals. In this section, details of analysis results regarding possible security risks and their possible solutions are given. The analysis results have been published in a scientific paper, namely "*Security Challenges of Location-based Mobile Business Applications*" [189]. As Table 3.1 shows, some challenges are common for each principal, whereas others are relevant only for a particular principal.

Assuming the infrastructure providers to be regarded as untrustworthy by all main principals and end-to-end security is enforced between the principals; we will not consider infrastructure providers in further detail.

### 3.2.1 Content and Communication Anonymity

Anonymity ensures that a user may use a resource or service without disclosing his/her real-world identity [77]. Similarly to non-electronic business, most users do not like to unnecessarily reveal their identity when requesting a mobile business service. For example, a celebrity may not want others

#	Security Challenge	U	B	SP
1	Content and communication anonymity	+	-	-
2	Privacy of personal data	+	-	-
3	Location-based spamming	+	+	+
4	Integrity and authenticity of service descriptions and results	+	+	+
5	Authentication and authorization	+	+	+
6	Confidentiality of the communication	+	+	+
7	Confidentiality of locally stored data	+	-	-
8	Secure software development	+	+	+
9	Usability vs. Security	+	-	-
10	Secure mobile payment and fair exchange	+	+	+
11	Rogue access points and forged GPS-signals	+	-	-

Table 3.1: List of Security Challenges  
(U:User, B:Broker, SP:Service Provider, +:challenge, -:no challenge)

to know which film in which cinema he/she watches, but also less famous people may not want others to learn what kind of books they buy and read.

Anonymity can be grouped as *content* and *communication* anonymity. Content anonymity focuses on hiding identity at the application level, whereas communication anonymity is related to the network level. If your real name is known by service providers, your content anonymity is compromised and your anonymity fails. Attacking your communication anonymity, service providers can elicit your IP address and find out your real identity, your location, etc. In order to provide anonymity, specific requirements for both content and communication anonymity should be satisfied.

A *partial* solution to content anonymity is pseudonymity. Pseudonyms are faked names like nicknames. When communicating with service providers, users introduce themselves with their pseudonyms instead of their real identities. They can use even different pseudonyms for the same providers. Even if a client uses pseudonyms for receiving services, each provider is likely to obtain some partial information about the clients, e.g. their location at the time the client requested the service or the company he/she works for. Based on the gathered information, a provider may not be able to determine the user's identity on his/her own, but collaborating service providers who are able to link the pseudonyms could eventually deduce the client's identity

from the data they have collected.

A more *complete solution* to content anonymity problem is identity management, i.e. letting the user retain control over their personal confidential information [87]. Identity management enables users to limit the amount of personal information revealed to certain providers or in certain applications. If certain information is not required by service providers, but they ask for data revealing identity or a profile, an identity management system should prevent this and provide content anonymity. Some solutions exist already for identity management on mobile platforms as explained in detail in Section 5.3.1. However, context-aware applications have further requirements for mobile identity management. For example, P3P [51] enables trust management in web surfing and its integration into mobile identity management would be an enhancement. Similarly, blurring location information in levels (e.g. city name instead of GPS coordinates) would also improve anonymity and privacy. Considering the new requirements, we have contributed to mobile identity management for context-aware applications as explained in Section 5.3.

When users communicate directly with service providers, their personal information (IP address, country, operating system and much more) [9] are all released to service providers which can then use such information to identify users and to profile them. Mix-net [103] based solutions [32, 106, 163] exist (see Section 5.2.1) that support unlinkability of transactions [77] and thus provide communication anonymity. The anonymity level provided by the existing solutions can vary depending on various parameters such as mix number, time delay, dummy messages, etc. Additionally, in the M-Business framework different users can have varying levels of anonymity sensitivity (e.g. celebrity vs. normal user) and also different applications can require varying anonymity levels (e.g. restaurant finder vs. mobile dating) for the same user. Considering these new requirements, a policy-based dynamic anonymity architecture is proposed as explained in Section 5.2.

### 3.2.2 Privacy of Personal Data

Regardless of whether content and communication anonymity can be guaranteed in the framework, users are generally concerned about revealing personal data [105], even if service providers are in practice unable to reconstruct their identities from the information they receive. In addition to conventional attributes such as name, address, phone and credit card number, special interests etc., this is especially the case for context information such

as the user's location at a specific point in time. The location information is, however, an essential input for any location-based service; a mobile dating service, for example, is pointless if the clients are not willing to disclose their location and some of their personal data.

Privacy of personal data should be supported by identity management (i.e. before users release data) and trust management (i.e. after users release data).

For identity management, Jendricke et al. present an identity manager to control personal data sent from mobile devices through networks [122]. An identity manager provides an interface with which one creates different virtual identifications (IDs), i.e. pseudonyms, and binds a subset of his/her personal data to each ID. When communicating with a service provider, the user chooses an ID that is suitable for this particular type of communication. Before any personal data is sent to a service provider, the user is explicitly asked to allow the transmission. This is discussed in Section 5.3.1 in detail and the enhancements required for mobile identity management are explained in Section 5.3.2.

In most cases, identity managers can ensure that each provider gets just as much personal information as needed for supplying the service requested, but how can service providers be prevented from abusing the legally collected information?

Obviously, on the technical level, the framework cannot control the further usage of information, once it has been transmitted to service providers. Abuse of gathered personal information for profiling etc. has to be prohibited on the business level, e.g. by establishing a Privacy Management Code of Practice that is obligatory for all service providers registering with the broker [177]. Providers violating this code will be banned from the service repository and not be able to further advertise their services through the broker. Legally proving a code violation is supposedly difficult, but the framework could enforce the pressure on malicious service providers by keeping log-files of transaction data, by collecting and managing abuse complaints and by operating complaint-dedicated communication channels between clients and service providers.

This countermeasure can be extended further by the integration of P3P trust management within identity management. Service providers specify their machine-readable P3P privacy policy as stated to the broker. Mobile identity managers of users know user privacy preferences and retrieve the relevant P3P policy before the users communicate with the providers. They compare the policy and the preferences and warn the users in case any conflict exists. The details of P3P-integration are explained in Section 5.3.5

in detail.

### 3.2.3 Location-based Spamming

Spams are unsolicited messages, mostly in the form of commercial advertisements. Meanwhile, the productivity of many companies is more and more affected by the increasing number of spam e-mails since employees need to spend a considerable amount of time separating wanted messages from unwanted advertisements.

Although many Internet users feel annoyed by spam e-mails, graphical user interfaces on modern PCs are able to give an easy overview of the e-mail inbox, and deleting a single message usually only takes a mouse click.

In a mobile environment, however, small display sizes force the user's attention onto each message, and the restricted user interface requires more user interaction during browsing and deleting. Since mobile phones and handhelds are trusted devices for many people, receiving unwanted messages on these devices is perceived as a massive privacy violation [177, 109].

In order to prevent anonymous spamming from unauthenticated sources, sender authentication can be established. It is the case that even authenticated service providers that are legal members of the service repository could betray the user's trust, send unwanted messages along with requested services and abuse personal data in order to perform personalized and location-based spamming. A shoe store could for example send advertisements to its customers when they pass by the shop.

One way to prevent this type of spamming would be to allow only pull-services, i.e. any communication between clients and service providers has to be initiated by the client. However, many presumably valuable push services exist, such as location-based notes or mobile dating services, which would be excluded from the framework by this approach.

In addition to the methods discussed in the previous subsection, the broker should rather support black or white listing of particular service providers. Clients can submit black lists and white lists to the broker, which then executes the lists on the user's behalf by never recommending services providers on the user's black list and assuming those on the white list to be trusted.



### 3.2.4 Integrity and Authenticity of Service Descriptions and Results

Integrity protects against unauthorized modification of information [165]. The integrity of the service descriptions stored and transmitted by the broker are obviously very critical for users and service providers because they affect choices of users when they decide on a service to request. Service descriptions from an authentic broker that are modified by adversaries or come from a spurious broker embarrass users and endanger the businesses of service providers. Adversaries are especially interested in modifying information about price, location and quality in descriptions.

To forge service descriptions, attackers can use a number of different methods. They can alter the service descriptions of the authentic broker. This modification can be done when service descriptions are either on the communication channel or in the repository of the broker. Another method is that a spurious broker pretends to be an authentic broker and sends forged service descriptions to mobile users. Thus, the authenticity rating of messages can also be forged. Modified service descriptions can result in many serious effects. Users can be charged more money than the required cost or pressurized to accept a bad service, for example. Even worse, users can be directed to a faked service provider whose aim is only profiling their personal data and stealing their credit card numbers. Like forged service descriptions, service results which are sent by service providers to mobile clients as replies to service requests can be forged on communication channels.

Digital signatures can be applied as cryptographic methods for both integrity and authenticity which require checking for unauthorized modification of messages and verification of the origins of service descriptions and results, respectively. To enforce a digital signature scheme, the broker and service providers should hold a public and private key pair. The broker should sign its service descriptions with its private key and then distribute them. Service providers should apply the same method for their service results. Users should check the integrity of service descriptions and results with the public key of the broker and service providers, respectively. Digital signature solutions usually require a certificate management system to exist in the framework.

The limited memory and CPU power of mobile devices are, however, big challenges when verifying message signatures. The verification algorithm running on mobile devices should therefore be well optimized.

In addition to the protection of unauthorized modification of messages on the channel, data stored in the repository of a broker should obviously be

authentic. Service providers as possible adversaries can aim to surpass that of their competitor service providers by modifying the repository in such a way that their service descriptions become more appealing to users.

The solution to the need for repository integrity is the enforcing of authentication, authorization and intrusion detection mechanisms. Authentication enables only authenticated principals to access the repository. Authorization provides that authenticated principals can work only with the data that they are allowed to access. Intrusion detection tools like Tripwire [72] can check and audit modifications in the repository and notify administrators of altered data.

### 3.2.5 Authentication and Authorization

After the registration process, service providers can apply to the broker to update their service descriptions. To prevent unsanctioned modification of the repository, the broker should authenticate and authorize the service providers. Authenticated service providers are then allowed to modify only entries that they own. Service providers can also require authenticating genuine brokers as opposed to spurious brokers and therefore bi-directional authentication is enforced.

Users will want to authenticate service providers in order to protect their personal data from malicious adversaries that pretend to be service providers. Conversely, many service providers need to authenticate their clients, e.g. for accounting purposes.

Authentication can be enforced by three different methods: something you know (e.g. passwords), something you have (e.g. smart cards) or something you are (e.g. fingerprints). Since all three methods by themselves would provide only weak authentication, a combination of two methods (*two-factor authentication*) is commonly used. As an example, token-based authentication requires a combination of the methods *what you have* and *what you know*. In token-based systems, the user holds a tamper-proof card that periodically generates a new random token. The same token stream is also generated on the remote server. The server authenticates the user if and only if the user is able to present the currently valid token and a PIN [55].

Two-factor authentication is desirable from a security point of view, but requires additional infrastructure and in many cases limits usability and scalability of the system as the authenticated entity has to provide at least two pieces of information in each authentication process. A quite natural solution is therefore to combine two-factor authentication and single sign-

on mechanisms [110] in order to ensure the usability of the system. With single sign-on, the identity is initially proved to the single sign-on service, and subsequent authentications are performed against the sign-on service instead of the authenticated entity itself. Both for authenticating users and service providers, a single sign-on service could be integrated into the broker.

Another important aspect regarding authentication is that anonymity should not be eliminated by authentication if users wish to stay anonymous while being authenticated. That means it should be possible for the user to show his/her authenticity by proving a certain fact about him/herself, e.g. being a legal subscriber to a service, without revealing the user's identity to the service provider. Conventional techniques such as transmitting passwords or biometric information or identifying the user's smart card do not protect his/her anonymity. Cryptographic techniques based on cryptographic credentials and zero-knowledge proofs of knowledge [116] provide a solution to this problem: The authenticator *can verify* that the user actually is a legal subscriber, but *cannot learn anything else* about the user's identity.

For authorization, many solutions such as access control lists, certificate-based authorization (e.g. SPKI [58]) which binds access rights to public keys, role-based authorization etc. exist. In order to keep the security architecture open, the architecture should not be restricted to only a specific set of solutions, rather all solutions required by different services should be provided.

### 3.2.6 Confidentiality of the Communication

Communication messages transmitted among the framework principals contain sensitive information such as personal data, credit card numbers, location, queries of mobile users, registration data of providers, results from broker and service providers etc. Identity management enables users to control the personal data transmitted, but the disclosure of these sensitive information would not be difficult in mobile networks where data is transmitted over air and easily received by any mobile device. To prevent the unauthorized disclosure of data in messages (confidentiality), encrypted message which only the authorized parties are able to decrypt and read messages is required.

Many telecommunication technologies provide encryption mechanisms between sender and network bearer. As a result of not trusted infrastructure providers, end-to-end security which enables confidential message transmission between the principals (users-broker, users-providers, broker-providers)

should be enforced. For end-to-end security, an SSL (Secure Socket Layer) [117] based protocol can be implemented. Authentication of broker and service providers, on-the-fly generation of session keys and its wide deployment in the public domain are the main advantages of SSL. In the protocol, messages are encrypted with a symmetric key, but public key encryption is used for the session key exchange. Hence, SSL requires also a certificate management system.

Communication protocols often require additional data to be associated with encrypted payload messages, e.g. for routing purposes. This data needs to be publicly readable and therefore must not be encrypted, but its authenticity should be established in the same way as for the payload message. In this context, authenticated-encryption with associated-data (AEAD) schemes [164] could significantly speed up the communication compared to conventional methods, especially on low-capability mobile devices.

Web Service Security (WS-Security) [49], a group of communications protocols for applying security to web services, also provides secure conversations methods namely WS-SecureConversation [84]. Unlike SSL, WS-SecureConversation supports end-to-end encryption. For example, if a message needs to go through any number of intermediaries before reaching to the final receiver and each intermediary needs to check the full or partial content of the message, the sender can encrypt the message individually for each intermediary. Even different parts of messages can be encrypted for different intermediaries and authentication of multiple party identities is also possible. These are quite useful features for context-aware applications in which different users and providers need to interact together during a multi-party service delivery. Even though SSL requires a separate security context for each communication party (i.e. point-to-point security), we have integrated SSL within the M-Business security architecture. This is because WS-Security libraries are not currently available for mobile device platforms and need to process very complex and processor time-consuming XML-based messages.

### 3.2.7 Confidentiality of Locally Stored Data

In the mobile domain, where thefts of devices are very common [78], confidentiality is especially required for protecting data stored locally on mobile devices. Local data is sensitive, because it contains private information such as name, address, special interests and possibly even credit card numbers. To prevent thieves and other unauthorized users from reading the data, the mobile device needs to authenticate the user trying to access it. This can

be done by two-factor authentication (see Section 3.2.5), e.g. by fingerprint authentication in combination with a PIN. Even if thieves figure out the PIN by brute-forcing, they will not be able to circumvent fingerprint authentication, and the device will consequently not allow access to the data.

However, in many cases, it is possible to get around the access control of the operating system by simply removing memory cards from the device and plugging them into another system. Therefore, sensitive data should always be stored in encrypted form, preferably by password-based symmetric encryption, in which passwords are used to generate keys for the encryption and decryption operations.

Alternatively, public key encryption can be used. The mobile user encrypts his/her local data with his/her public key. The corresponding private key is stored by a remote system and can only be retrieved after authentication by a password.

### 3.2.8 Secure Software Development

Security is a non-functional property of applications. During the design phase of software developments, integration of security aspects is mostly not taken into consideration and is postponed to the implementation stage. Applications are implemented – and then it is considered how to secure the application. However, this approach does not work, since security requirements can conflict with the architecture design. This is one of the common mistakes during software development. Therefore, the need for stringent security should come into play during the initial design phase.

Another dilemma of software development regarding non-functional properties is bad modularity due to crosscutting concerns (e.g. security, exception handling, logging, database transactions, etc.). Security concerns cross-cut with other application modules and this causes decomposition problems [201, 200, 195]. For better management of security during software development and execution phases, separation of business and security logics is required [198, 199, 182].

Managing security at the design phase and separating business and security logics are good approaches for enhanced security. However, implementing applications with bugs is another risk. Considering bugs that are related to security modules, the risks are higher and more dangerous. A small security bug may cause failure of the business itself. Mostly, designs and implementations might be correct but their implementations contain bugs due to human factors. If generation of source code from specifications can be realized automatically, the risks from buggy implementations can be

minimized (see Chapter 6).

Software bugs in mobile clients, broker and service provider applications should be consequently non-existent. Hence, security components should be designed and integrated into the software during the application design phase. Separation of security from business logic is provided as far as possible and also automatic generation of final source code according to security protocol specifications are also provided for the M-Business framework.

### 3.2.9 Usability vs. Security

It is a widely known fact that users, when faced with trading-off usability and security, mostly prefer usability. As an example, consider password-based authentication of service subscribers. To ensure the security of the authentication, passwords must not be easily guessable, i.e. they should not be chosen from dictionaries nor should they be names or birth dates. Instead, passwords should contain capital letters, numbers and even non-ASCII characters. Strong passwords increase security but they are not easy to keep in mind and thus decrease usability. In spite of feeling uncomfortable about it, many people nevertheless use weak passwords.

Another trade-off example is digital certificates. When the lifetime of a certificate is over, it no longer guarantees the authenticity and validity of its content. When a mobile device receives an invalid certificate from a service provider or the broker, it should warn the user in a suitable manner. Users, however, have different sensitivity regarding security. While invalid certificate warnings are annoying and therefore decrease usability for some users, others may find such warnings inevitable and desirable.

As both examples show, the M-Business framework should allow users to balance usability and security according to their personal needs and not enforce fixed security policies.

A dynamically configurable policy-based security management system is a possible solution. Such a security management system can consist of the following components and mechanisms:

- *Password Manager*: A password manager creates strong passwords for different services, and encrypts and stores them on the local storage medium. Users then do not need to worry about remembering all strong passwords or using weak passwords. They only have to keep in mind a master password for authentication by the password manager and to retrieve passwords at any time.

- *Single-Sign-On (SSO) Mechanism*: With the help of SSO, not all service providers need to authenticate a particular user. Instead, a central authentication server performs this task on behalf of the service providers.
- *Security Level Manager*: The security level manager presents different security levels (e.g. high, medium, low), and each level is bound to a set of security options. Users can easily and dynamically switch from one security level to another and also enable or disable any option individually for each level.
- *Identity Manager*: An identity manager, as explained in Section 5.3, provides full control over the disclosure of personal data in each transaction and therefore increases usability as well as security. Since location information is generally considered to be very sensitive, the client could trade-off security and quality of the returned service by adjusting the accuracy of the location information that is transmitted to the service provider, e.g. instead of transmitting exact GPS-coordinates, the mobile client could send only the district or even only the city that he/she is currently located in.

### 3.2.10 Secure Mobile Payment and Fair Exchange

Mobile payments involve transactions in which monetary values are transferred from mobile clients to service providers in order to pay for services offered. Suitable monetary values for mobile payments are digital coins, which can be stored on either the mobile device itself or smart card devices (e.g. German Geldkarte). Alternatively, monetary values stored with a remote trusted party (e.g. a credit card or an account in a bank) can be also used for payments.

Credit card numbers and other payment media can be stolen during message transmissions of payment protocol and misused by criminals. Misuse can be a serious matter for both users and service providers. It gives difficulties users, because their credit cards were used on their behalf. It also is detrimental to service providers, because in the case the users claim that they were charged due to misuse of their credit cards, the money can possibly be refunded by the providers. Hence, the mobile payment protocol (or protocols) that would be deployed in the M-Business framework should consider strong encryption methods to provide confidentiality of monetary values transmitted over unreliable networks.

Mobile users expect service providers to play fair when they exchange service and monetary values. Otherwise, they may complain about an unfair exchange and argue that:

- They were charged more money than the expected value.
- They were charged more than once.
- They were charged even though they did not get a service.
- They were charged even though they got a wrong service.
- They were charged even though they did not request any service.
- They were charged even though they are unhappy with the quality of the received service.

Similarly, service providers also expect mobile users to play fair when they exchange services and monetary values. Service providers can also have different arguments to claim against users, e.g., that they are not paid although they presented a service, they are not paid in time, or they are paid less than the agreed amount.

In payment schemes, if any dispute comes up between client and merchant, they need a trusted third party to solve the dispute. In the M-Business framework, the broker can take the role of such a trusted party.

In order to provide evidence in cases of dispute, the payment protocols should be able to account for all transactions of both parties. They should also provide anonymous payment for certain applications, accountability (*non-repudiation*), for users and service providers as well as mechanisms verifying the authenticity and integrity of protocol messages.

### 3.2.11 Rogue Access Points and forged GPS-signals

Access points that are illegally attached to WLAN networks are called rogue access points. If a rogue access point is attached to the M-Business framework, mobile devices may fail as regards location determination, or sensitive user data would be transmitted over rogue access points, which is dangerous in terms of protection of privacy.

To prevent illegal attachments of access points, infrastructure providers should make regular checks in order to detect rogue access points. Common detection techniques are based on only wireless (e.g. sniffers for packet



analyzing, enterprise-wide scan from a central location), only wired (e.g. MAC address filtering) and hybrid approaches [98].

As well as access points, GPS-satellites can be fraudulently simulated by ground stations transmitting GPS-signals in order to falsify the location determination of clients. Since authentication of civilian GPS-messages is not yet available [202] and will only be included in next-generation location determination systems like GPS III and Galileo [152], particularly critical applications should not trust the location information provided by GPS.

### 3.3 Extra Limitations

Since security issues directly affect the user-acceptance of M-Business applications, they are among the most important challenges for M-Business framework. However, for several reasons, providing security is very problematic and difficult:

- *Security is a difficult challenge in general:* Providing security is an engineering task. A “normal” engineer specifies the challenges and provides solutions. A security engineer, however, tries to win a game against a dynamically evolving malicious adversary. While the challenges that a normal engineer faces usually do not change unexpectedly, this is a quite normal and common situation for a security engineer.
- *There exists a trade-off between functionality and security:* System designers always face a trade-off between functionality and security. Since security is a non-functional aspect of a system and end users often intuitively prefer increased functionality over enhanced security, this leads to overlooked security challenges.
- *There are additional security challenges for mobile systems:* Limited capabilities of mobile devices prevent the deployment of common security solutions in the mobile domain. As an example, signing documents with digital signatures in order to ensure integrity in many cases requires much CPU power, and not all mobile devices are capable of completing this task. Similarly, limited I/O functionality prevents long passphrases and other advanced security-related user interaction. Another challenge comes from wireless communication. It is obviously much easier to eavesdrop data that is transmitted over the air than to intercept wired communication channels. Also, it is

much more difficult to detect a wireless eavesdropper than to detect that someone has hooked into a wired connection. In addition, the mobile communication environment changes steadily and there is no implicit authentication by “being connected to the cable”.

- *Support for security in standards is marginal and often broken:* Typically, standards for wireless and mobile communication only provide support for basic security features, such as confidential and authentic communication between a mobile device and the next base station or access point. Advanced security features thus have to be implemented at the application level. However, even those basic security features actually supported by standards often are broken by design. A well-known example is “Wired Equivalent Privacy” (WEP) from the IEEE 802.11 standard [197]. (The worst flaws of WEP are fixed now by “WiFi Protected Access” (WPA) and by IEEE 802.11i, finally approved in July, 2004.) Other old and well-known examples are due to the GSM mobile standard: The one-sided authentication protocol (only the mobile device authenticates itself to the base station, but the base station does not authenticate itself to the mobile device), and the insecurity of the GSM A5 stream cipher [204].
- *A new privacy challenge is how to control location information:* This challenge stems from unauthorized disclosure of location information. The owner of a device should be able to explicitly control the transmission of his/her location. If the location becomes available to malicious adversaries, privacy issues — possibly as severe as danger to life and limb — may arise.

## Chapter 4

# Exploits against User Privacy

### 4.1 Privacy Challenges

In 1999, the chief executive officer of Sun Microsystems Scott McNealy said that “You have zero privacy anyway. Get over it.” [179]. Today in the Internet era, privacy has become a more critical issue. As summarized very well in Figure 4.1, much personal data are shared over many different applications. Being concerned with emails, online banking, e-government applications, online groups and communities, e-commerce, social networks, blogs, etc., protecting privacy has become nearly impossible. Nevertheless, we will take into consideration all social, legal and technical aspects of privacy and propose solutions to help individuals to safeguard their privacy against the personal data threats.

Privacy risks go beyond the mere collecting of some personal data. Violations can even threaten the lives of people in certain circumstances. People might not be aware of the importance of their privacy unless they are threatened by exploitation of their personal data. People would become more sensitive in terms of their privacy if they are informed about the real-life abuses of privacy and its (often serious) consequences. If an attacker (i.e. a person or a service provider) retrieves your personal data and misuses it for malicious purposes such as profiling, spamming or selling to third parties, you would be more conscious about the need for the protection of your privacy.

In context-aware applications, service providers and other principals have also great opportunity if they want to threaten user privacy [142]. In this

section, some examples of a number of privacy violations against mobile users in the Friend Finder application are given. The possible *attackers* are other users, the location provider or other third party providers.

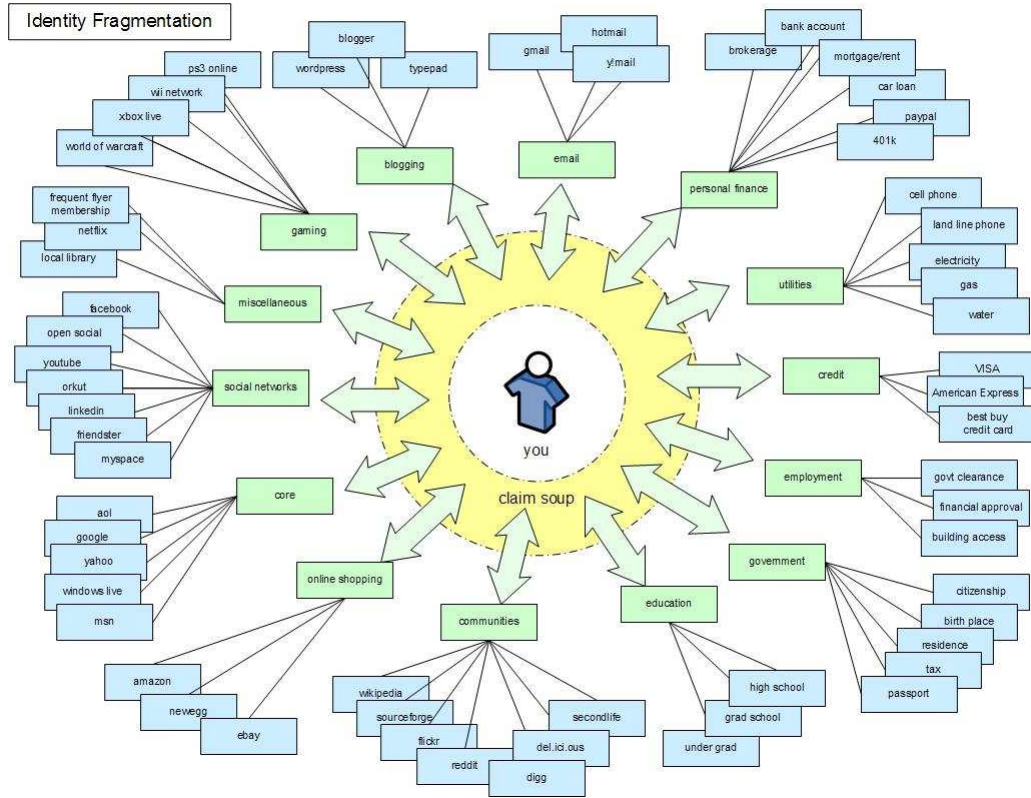


Figure 4.1: Fragments of Identity<sup>1</sup>

#### 4.1.1 Location- and Action-relevant Risks

Considering a “Big Brother”-like life, most people would not prefer that others know exactly what they are doing and where they are going during a day.

This risk exists in the Friend Finder application. With the aim of tracking users and revealing their actions and activities, attackers can collect and analyze location data of users. They can even target a single user and store his/her location information at different times at different places. They can

<sup>1</sup>Referenced from <http://www.identityblog.com/?p=893>

even find out where the mobile user was (e.g. if he/she was in a night club or fitness center), follow his/her actions or even guess the places at which he/she would be in the future.

Additionally, not only the providers but also other mobile users may want to reveal the location of a particular user without having any permission. As an example, your boss can threaten your privacy by trying to find out where you go and what you do in the evenings. This information is private to you and should also remain confidential.

### 4.1.2 Relationship-relevant Risks

As a member of the society, people can build contacts with many people (i.e. friend, relatives, family, etc.). That means each person has his or her own social network which is private to himself or herself. People do not need or want necessarily to release their contact lists to others. The existing social networks (e.g. Facebook [17], Xing [76]) support enabling or disabling of contact list release to others as a privacy preference.

This privacy risk exists also in the Friend Finder application. With the aim of revealing the relations among mobile users, attackers collect location information and try to find out who stay in the same place or travel towards the same direction at the same time. This would give hints about the relationships among certain users. Additionally, the friend-search queries can be collected and analyzed in order to reveal communications and relationships between different people.

### 4.1.3 Monetary Risks

In actual, physical life, many people prefer hiding certain personal features like wealth, i.e. having something that is very costly. Mobile devices which contain advanced functions and thus are very expensive would seem to be ideal candidates for thieves to steal. Therefore, certain aspects of mobile devices are private and should be kept private. In the Friend Finder application, the risks exist that the private aspects of mobile devices and users themselves can be retrieved by potential and actual attackers.

The User Agent Profile (UAProf) [73] is a specification for capturing capability and preference information for wireless devices. Content providers can benefit from this information by creating content in an appropriate format for the specific device [69]. Device capabilities (i.e. device model) can make it clear, for example, whether it is a cheap or costly device [148, 111].

Retrieval UAProf data by attackers can result into risk from two different

perspectives. Firstly, thieves can locate costly devices and target their owners to steal their devices. Secondly, the location provider can collect UAProf data sent by mobile users and profile them according to their mobile device types. The profiles can be shared with third-party mobile device companies to send unsolicited advertisements to mobile users. This spamming problem would threaten the privacy of mobile users.

#### 4.1.4 Medical Data Risks

Privacy of medical data is most desirable for nearly all people. Revealing that what illnesses or handicaps you have is not something you would prefer. A concrete example has been experienced in Germany in August 2008 (see Section 4.3). The German health insurance organization DAK forwarded sensitive health data of over 200.000 members illegally to a private US-company (namely Healthways) without getting the consent of these members [12]. As a result, Krankenkasse members/contributors have become very anxious about their privacy.

However, UAProf specifications contain preference information of wireless devices, too. User preferences (e.g. font size) can give hints about users' visual acuity. People with weak visual acuity prefer big font sizes and displays with image-disabled functionality, for example.

#### 4.1.5 Dynamic Pricing

The price of a service with a certain quality is expected to be a fixed value. In contrast service providers may require different payment amounts for the same service based on the wealth of a person or their nationality. For example, a payment policy such as "the price is 10 Euros higher for people from European countries" or another policy "rich people pay more" can easily be applied. This is called *dynamic pricing*.

Dynamic pricing is also another monetary risk in the Friend Finder. The location provider can analyze how frequently mobile users retrieve the service and make profiles of users based on their purchasing activities. Afterwards, they can apply dynamic pricing. In addition, UAProf profiles the owners of costly devices can also help attackers/commercial enterprises to apply dynamic pricing. These privacy risks ought to be avoided in the Friend Finder application.

## 4.2 Legal Directives

Considering privacy as a human right from the legal perspectives, the European Parliament and the Council of the European Union have published the directive 95/46/EC [16] and the directive 2002/58/EC [15] as a continuation of the directive 95/46/EC.

Considering the legal aspects, the privacy and security threats should be carefully taken into consideration when technical systems are designed and implemented. This requires service providers and the broker to take heed of privacy and security risks and system developers to develop and integrate PET (privacy enhancing technologies) tools within software architectures in order to support mobile users.

### 4.2.1 The Directive 95/46/EC (Data Protection)

The directive 95/46/EC (*in full title “Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data”*) published in 1995 focuses on data protection.

The object of the directive is defined in Article 1:

*“In accordance with this Directive, Member States shall protect the fundamental rights and freedoms of natural persons, and in particular their right to privacy with respect to the processing of personal data.”*

The definition of personal data is given in Article 2-a:

*“personal data shall mean any information relating to an identified or identifiable natural person (data subject); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his/her physical, physiological, mental, economic, cultural or social identity”*

Article 7-a requires that personal data of a data subject can be only processed if he or she has given his/her consent:

*“Member States shall provide that personal data may be processed only if:*

*(a) the data subject has unambiguously given his consent, or”*

Privacy of personal data is guaranteed in Article 8-a, but 8-b and 8-c detail some exceptions:

*“Member States shall prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life.”*

The rules of data collection are explained in Article 10:

*“Member States shall provide that the controller or his representative must provide a data subject from whom data relating to himself are collected with at least the following information, except where he already has it:*

*(a) the identity of the controller and of his representative, if any;*

*(b) the purposes of the processing for which the data are intended;*

*(c) any further information (such as the recipients of the data, whether replies to the questions are obligatory or voluntary, the existence of the right of access to and the right to rectify the data concerning him)”*

In this directive, it should be understood that brokers and service providers must obey the rules in order to help individuals to guarantee their personal data privacy.

#### 4.2.2 The Directive 2002/58/EC (E-Privacy)

This directive is a complement to the EU Directive 95/46/EC. It is concerned with the processing of personal data and protection of privacy in the electronic communication sector.

Relevant to context-aware and location-aware applications, it addresses issues such as security, confidentiality, data storage and location data.

According to Article 4 (*Security*); service providers must take appropriate measures to safeguard the security of their services. If a security risk exists, their users should be informed of this risk and any likely costs involved with providing the possible remedies.



According to Article 5 and 6 (*Confidentiality*); member states shall ensure the confidentiality of communications and the related traffic data through national legislation. They shall ensure also that access to and processing of the data is allowed only if the user concerned is clearly informed and gives his consent.

According to Article 6 (*Data Storage*); user data can be stored and processed by service providers only for the duration necessary for the services and billing purpose. Service providers should always keep subscribers informed of the types of data they are processing and the purposes and duration for which this is done. The data stored must be also erased or made anonymous when it is no longer needed for the purpose of the transmission. In addition, the users should be always in the position of withdrawing their consents to store and process their data.

According to Article 7 (*Itemized Bills*); itemized bills improve possibilities for the subscribers to check the accuracy of the fees charged by the service providers but at the same time it may jeopardize the privacy of the users of publicly available electronic communications services. Therefore, subscribers should have the option of receiving non-itemized bills or privacy enhancing methods of communication and payments.

According to Article 2 and 9 (*Location Data*); location data means any data processed in an electronic communications network, indicating the geographic position of the terminal equipment of a user of a publicly available electronic communications service. Service providers must inform their users or subscribers, prior to obtaining their consent, of the type of location data other than traffic data which will be processed, of the purposes and duration of the processing and whether the data will be transmitted to a third party for the purpose of providing a value added service. Users or subscribers shall be given the possibility to withdraw their consent for the processing of location data other than traffic data at any time.

In this directive, privacy of location data is explicitly protected. Brokers and service providers collecting location data of users must consider these legacy rules, define their privacy policies accordingly and enforce their privacy practices as stated in the EU directives.

### 4.3 Privacy Threats in the Media

Much news is published in the media about threats to privacy. People are theoretically at risk by using online banking, publishing private information in blogs or social networking sides, etc. In this section, some concrete exam-

ples are given to show that privacy can be a headache in cases where privacy is not seriously considered and the required measures/countermeasures are not taken.

In early 2000s, Ellen Simonetti became a victim of a privacy issue [176, 192]. She was a flight attendant for Delta Air Lines for 8 years, but was dismissed from her job since she had published her “*inappropriate*” pictures in Delta uniform on her personal web blog [8].

Identity theft in online banking is also a very critical issue in terms of privacy. Applying phishing attacks, attackers steal and misuse identities (e.g. passwords, bank account data, PIN numbers, TAN numbers) of victims and execute e-banking transactions on behalf of the victims. Many real life identity theft events have occurred [11] and the victims have been left in desperate situations.

A serious privacy violation happened in August 2008 in Germany, as mentioned in the previous section. It was realized that the German health insurance provider (Krankenkasse) DAK had forwarded sensitive health data of over 200.000 patients illegally to a private US-company (namely Healthways) without getting the consent of their members [12]. The DAK has been criticized very extensively for causing this privacy violation and their members (i.e. the victims) have become very anxious about their privacy.

Some shopping companies started tracking customers by listening on the whisperings of their mobile phones [149]. They can tell when people enter a shopping center, what stores they visit, how long they remain there, and what route they took as they walked around.

Mobile phones can be further used to track anyone thanks to a service called World Tracker which lets you use data from cell phone towers and GPS systems to pinpoint anyone’s exact whereabouts, any time as long as they have their phone on them. The service shows you the exact location of the phone by the minute, conveniently pinpointed on a Google Map [162].

The anonymity network Tor has been also targeted for stealing usernames and passwords [52]. The exit nodes can access unencrypted data if no end-to-end encryption (i.e. SSL, TLS, HTTPS) is enforced between the sender and the receiver. The hacker Dan Egerstad equipped with 5 exit nodes in the Tor network and sniffed 100 log-in credentials of mainly embassies and consulates in different countries. He publicized the list of the credentials, but then was arrested by the Swedish police [172].

Attackers can find their victims today with Google search engine (see Section 4.4). This is called as Google hacking and much news has already appeared in the media regarding the risks of Google hacking [139]. The following examples show the revealing of private information by using Google.

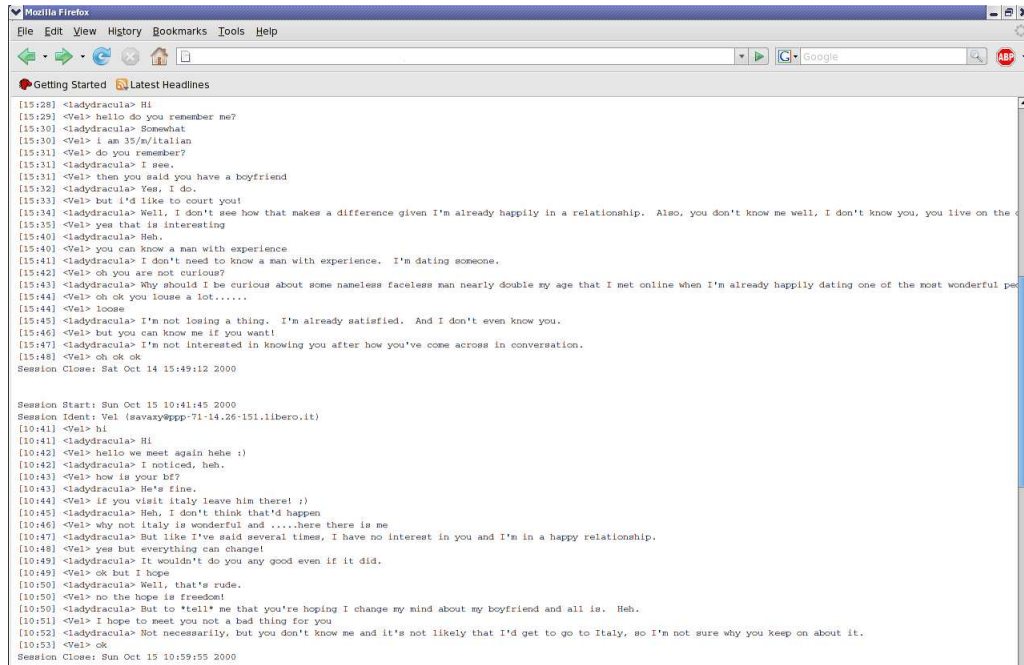
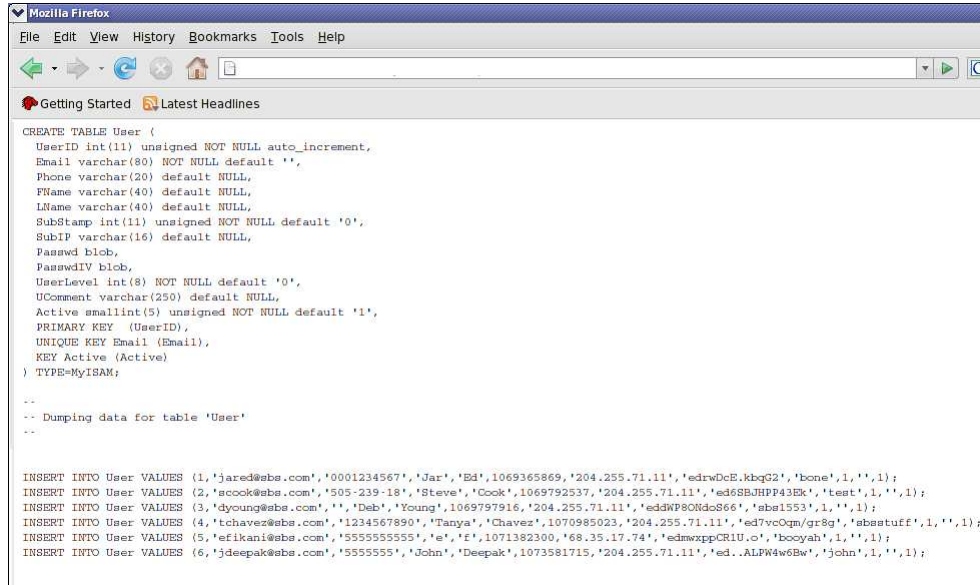


Figure 4.2: A private Chat Log

In Figure 4.2, the private chat logs between an Italian man (age 35) and a younger girl can be followed. In Figure 4.3, passwords are revealed from a dumped database. In Figure 4.4, the private inbox and sent emails of an online journal are listed. In Figure 4.5, many confidential documents are listed by Google.

Another critical tool in terms of privacy of Google is Google Calendar. Attackers can use it for searching for private personal data (e.g. appointments, credit card numbers). Even sensitive business data can be made public on the Google calendar [140]. As concrete examples, McKinsey dial-in information was posted by an employee who had shared some calendar events including project status meetings and call-in numbers for a company project. This was also confirmed by the spokesman of the company. In addition, the details for several JPMorgan Chase & Co. conference calls relating to the company's storage systems, including a dial-in number and passcode for a telephone call could be seen publicly.

Google Earth can be also used for privacy threats. For example, UK-Teens have used Google Earth images to find houses with swimming pools.



```

CREATE TABLE User (
  UserID int(11) unsigned NOT NULL auto_increment,
  Email varchar(80) NOT NULL default '',
  Phone varchar(20) default NULL,
  FName varchar(40) default NULL,
  LName varchar(40) default NULL,
  SubStamp int(11) unsigned NOT NULL default '0',
  SubIP varchar(16) default NULL,
  Password blob,
  PasswordIV blob,
  UserLevel int(8) NOT NULL default '0',
  UComment varchar(250) default NULL,
  Active smallint(5) unsigned NOT NULL default '1',
  PRIMARY KEY (UserID),
  UNIQUE KEY Email (Email),
  KEY Active (Active)
) TYPE=MyISAM;

--
-- Dumping data for table 'User'
--

INSERT INTO User VALUES (1,'jared@sbs.com','0001234567','Jar','Ed',1069365869,'204.255.71.11','edrwDcE.kbqG2','bone',1,'',1);
INSERT INTO User VALUES (2,'scook@sbs.com','505-239-18','Steve','Cook',1069792337,'204.255.71.11','ed6SBjHPP43Er','test',1,'',1);
INSERT INTO User VALUES (3,'dyoung@sbs.com','','Deb','Young',1069797916,'204.255.71.11','ed3WP80Kda866','sbs1553',1,'',1);
INSERT INTO User VALUES (4,'tchavez@sbs.com','1234567890','Tanya','Chavez',1070985023,'204.255.71.11','ed7vcOqm/gr8g','sbsstuff',1,'',1);
INSERT INTO User VALUES (5,'efikani@sbs.com','5555555555','e','f',1071382300,'68.35.17.74','edmwpxpCR1U.o','booyah',1,'',1);
INSERT INTO User VALUES (6,'jdeepak@sbs.com','5555555','John','Deepak',1073581715,'204.255.71.11','ed..ALPW4w6Bw','john',1,'',1);

```

Figure 4.3: Dumped Passwords

Once they find a target, they use Facebook to arrange an organized, but uninvited pool-crash [175].

All these *real-life* privacy exploits in this section should have shown that privacy can be threatened any time, anywhere, even by people or organizations with which you are not familiar. It is therefore highly desirable that individuals should be aware of possible threats, take the required counter-measures and consequently protect themselves.

## 4.4 A Case Study: Google Hacking

This subsection focuses on a case study for privacy exploits, i.e. Google Hacking and some real life attacks against user privacy are presented. The exploits are not applied directly in context-aware applications, but they make it clear that if users do not take into consideration the necessity of protecting themselves against possible violations of their privacy, they most probably would fail to enable a secure and private digital life for themselves in any kind of application.

The exploitations of privacy violations are based on web search engines which are the biggest service providers (e.g. Google, Yahoo, Lycos, etc.)

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	24-Aug-2007 00:38	-	
<a href="#">Advertising.dbx</a>	02-May-2004 23:28	198k	
<a href="#">Deleted Items.dbx</a>	02-May-2004 23:28	136k	
<a href="#">Drafts.dbx</a>	02-May-2004 23:28	136k	
<a href="#">Editorial Staff.dbx</a>	02-May-2004 23:28	520k	
<a href="#">FAQ.ed03.com.dbx</a>	02-May-2004 23:28	59k	
<a href="#">Folders.dbx</a>	02-May-2004 23:28	73k	
<a href="#">HD.dbx</a>	02-May-2004 23:28	136k	
<a href="#">Inbox.dbx</a>	02-May-2004 23:28	2.1M	
<a href="#">MindMagazine.dbx</a>	02-May-2004 23:29	2.1M	
<a href="#">NobleHosts.com*.dbx</a>	02-May-2004 23:29	1.0M	
<a href="#">Offline.dbx</a>	02-May-2004 23:29	9k	
<a href="#">Outbox.dbx</a>	02-May-2004 23:29	187k	
<a href="#">Pop3uid1.dbx</a>	02-May-2004 23:29	9k	
<a href="#">Sent Items.dbx</a>	02-May-2004 23:35	28.9M	
<a href="#">Web Hosting Review.dbx</a>	02-May-2004 23:35	136k	
<a href="#">cleanup.log</a>	02-May-2004 23:28	37k	
<a href="#">iQuality Award.dbx</a>	02-May-2004 23:29	1.2M	
<a href="#">off-topic.ed03.com.dbx</a>	02-May-2004 23:29	59k	
<a href="#">support.ed03.com.dbx</a>	02-May-2004 23:35	59k	
<a href="#">webdev.ed03.com.dbx</a>	02-May-2004 23:35	59k	

Apache/1.3.37 Server at Port 80

Figure 4.4: Private Emails

for information searches in the Internet. However, they threaten personal privacy by indexing more and more secret and private data for unauthorized access. The biggest threats can result from their “indexing anything” features. In particular, Google is the most popular web search engine on the Internet. It indexes vast amounts of information from web servers thanks to its hardworking web crawlers. As a result, sensitive personal data that should be kept secret and confidential are indexed by Google, too. Personal data like name, address, phone numbers, emails, CVs, chat logs, forum and mailing list postings, username-password pairs for login sites, private directories, documents, images, online devices like web cameras without any access control, secret keys, private keys, encrypted messages, etc. are all available to others via Google. This is called *Google Hacking* and threatens

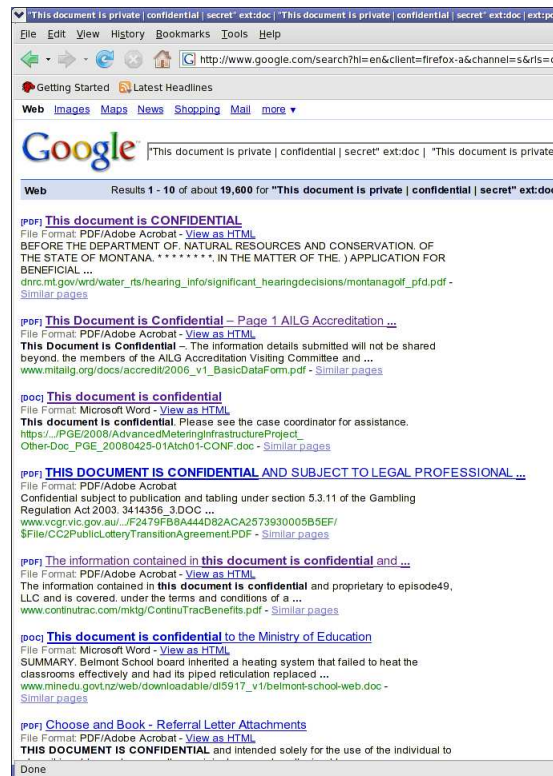


Figure 4.5: Confidential Documents

our privacy. In addition to the privacy risks, there might exist other security threats that can be revealed by Google. There is an online database [25], which contains 1423 different Google hacking search queries in different categories (e.g. files containing juicy info, pages containing login portals, various online devices, vulnerable servers, etc.) as at November 2008.

In addition, attackers can use automatic tools to execute attacks on privacy and reveal sensitive, confidential and secret personal data. User-centric countermeasures should be applied by individual users to safeguard against Google Hacking.

#### 4.4.1 Google Search Parameters

In addition to the basic search operators (i.e. +, -, .), Google supports other parameters (i.e. intitle, inurl, intext, filetype, site, etc.) for advanced

searches and filters its results according to these parameters provided by users.

“+”, “-”, “\*”, “|” and quotation (“ ”) are the basic search operators. To retrieve particular pages containing some certain keywords, the plus (+) operator can be used to combine the keywords. The minus (-) operator is placed just before a keyword and pages containing this keyword are removed from the result list. It can be put also in front of an advanced operator and reverses its behavior. As an example, a search query containing the parameter *-site:www.example.com* will not list the results from *www.example.com*. The star (\*) operator is used as a wildcard operator. The operator “|” or the keyword “OR” can be used for combining different search queries with a *logical OR*. Keywords put within quotations (“ ”) are searched for as a phrase.

Advanced operators related to Google hacking are *inurl*, *intitle*, *intext*, *site*, *filetype* and *ext* [23, 134]. *[all/inurl]* parameter is used to filter out the results according to a certain keyword contained in url. If more than one keyword is needed for filtering, the *allinurl* parameter should be used. *[all/intitle]* filters the results according to titles of web pages. *[all/intext]* searches the keywords in the body of web pages. With the parameter *site* you can apply host-specific searches. *filetype* and *ext* parameters have the same functionality and are needed to filter out the results based on file extensions (e.g. *html*, *php*, *pdf*, *doc*, etc.).

A concrete example can help to understand these parameters. For example, you are interested in finding *security*-relevant *pdf* documents in *Germany*. The query (*intitle="security" ext:pdf site:de*) can precisely define your search and return better results from Google.

#### 4.4.2 Exploits against Personal Data

Google can be queried for revealing sensitive personal data by using its advanced search parameters [186, 187]. An attacker can target a certain person and benefit from automated tools exploiting the person’s privacy.

The private data searches are grouped into four different groups according to the privacy level. These are *identification* data, *sensitive* data, *confidential* data and *secret* data searches.

##### Identification Data

The identification data is related to personal identity of users. Name, surname, address, phone number, marital status, CV, alias, nickname used over

the Internet, etc. are the typical examples of the identification data. Some private data searches focus on a certain person and the most common name “Thomas Fischer” in Germany is chosen as the test candidate<sup>2</sup>.

*Name, Address, Phone, etc.*

You can search for web pages and documents which contain keywords like name, surname, address, phone numbers, birthday, email, etc., optionally for a certain person or within certain document types.

```
allintext:name email phone address intext:"thomas fischer" ext:pdf
```

TWiki<sup>3</sup> is a wiki-based web application that is commonly used for project management. Inside TWiki, user data like name, address, phone numbers, web pages, location, emails, etc. are stored. If the required authentication techniques are not enforced, unauthorized people can also access this data.

```
intitle:Twiki inurl:view/Main "thomas fischer"
```

In addition to Google search, other search engines with the “people-find” capability can also be very helpful for obtaining identification data. Yahoo’s People Search<sup>4</sup>, Lycos’s WhoWhere People Search<sup>5</sup> or eMailman’s People Search<sup>6</sup> connecting public ldap servers are examples of such services. Similarly, the Firefox plug-in “People Search and Public Record Toolbar”<sup>7</sup> gives you many facilities to search for the identification data.

*Curriculum Vitae*

You can search for the keyword CV (curriculum vitae) that return documents containing identification data. This search can be extended by searching for “CV” in different languages. For example, *Lebenslauf* can be used within the search query as the German translation for CV.

---

<sup>2</sup>We have come to this conclusion by comparing different combinations via [www.googlefight.com](http://www.googlefight.com)

<sup>3</sup>TWiki: <http://twiki.org>

<sup>4</sup>Yahoo People Search: <http://people.yahoo.com>

<sup>5</sup>Lycos People Search: <http://peoplesearch.lycos.com>

<sup>6</sup>eMailman People Search: <http://www.emailman.com/ldap/public.html>

<sup>7</sup>People Search and Public Record Toolbar, <https://addons.mozilla.org/en-US/firefox/addon/3167>



```
intitle:CV OR intitle:Lebenslauf "thomas fischer"  
intitle:CV OR intitle:Lebenslauf ext:pdf OR ext:doc
```

### *Login Names*

The Webalizer application<sup>8</sup> collects statistical information over web sites about their visitor activities. The most commonly used login names are also stored by Webalizer.

```
intitle:"Usage Statistics for" intext:"Total Unique Usernames"
```

### **Sensitive Data**

With sensitive data, it is meant that data which is normally public but its revelation may disturb its owner in certain situations. Examples are postings sent to forums, emails sent to mailing lists, sensitive directories and Web2.0-based social networking applications.

### *Forum Postings, Mailinglists*

PhpBB<sup>9</sup> is a widespread web forum application. It enables the collecting of all postings sent by a particular user. The following search finds out all postings sent with the alias thomas to different phpBB-based forums.

```
inurl:"search.php?search_author=thomas"
```

Mailman<sup>10</sup> is a well-known mailing list manager. The following search gives all email postings which are sent to mailman-based lists and related to *Thomas Fischer*.

```
inurl:pipermail "thomas fischer"
```

### *Sensitive Directories*

Backup directories can contain sensitive data about users, organizations, companies, etc.

---

<sup>8</sup>Webalizer: <http://www.mrunix.net/webalizer/>

<sup>9</sup>PhpBB Forum: <http://www.phpbb.com>

<sup>10</sup>Mailman List Manager: <http://www.gnu.org/software/mailman/>

```
intitle:"index of" inurl:/backup
```

### *Web2.0 Applications*

The next generation Web2.0 applications introduce more privacy risks. People share more personal data with others within Web2.0-based social networking and blogging applications. The following searches are based on the favorite Web2.0 services like Yahoo's Image Sharing<sup>11</sup>, Google's Blogger<sup>12</sup>, Google's Video Sharing<sup>13</sup> and Facebook Social Networking<sup>14</sup>. Instead of searching through Google, searching directly on the original sites would give more efficient results.

```
"Thomas Fischer" site:blogspot.com  
"thomas" site:flickr.com OR site:youtube.com  
"thomas fischer" site:facebook.com
```

### **Confidential Data**

The confidential data is normally expected to be non-public for others except for a group of certain people, but Google makes it possible to access such private data as well.

### *Chat Logs*

You can search for chat log files related to a certain nickname.

```
"session start" "session ident" thomas ext:txt
```

### *Username and Password*

Username-password pairs can be searched within sql dump files and other documents.

```
"create table" "insert into" "pass|passwd|password" (ext:sql  
| ext:dump | ext:dmp | ext:txt)
```

---

<sup>11</sup>Yahoo Image Sharing: <http://www.flickr.com>

<sup>12</sup>Google's Blogger: <http://www.blogspot.com>

<sup>13</sup>Google Video Sharing: <http://www.youtube.com>

<sup>14</sup>Facebook-Social Networking: <http://www.facebook.com>

```
"your password is *" (ext:csv | ext:doc | ext:txt)
```

#### *Private Emails*

Microsoft Outlook and Outlook Express store personal emails in single files such as incoming messages “inbox.dbx”. The following searches target the email files stored by Outlook Express or Microsoft Outlook.

```
"index of" inbox.dbx  
"To parent directory" inurl:"Identities"
```

#### *Confidential Directories and Files*

Confidential directories and files can be revealed with the following query.

```
"index of" (private | privat | secure | geheim | gizli)
```

In order to prevent web crawlers listing private directories, Robot Exclusion Standard [54] is used.

```
inurl:"robots.txt" "User-agent" ext:txt
```

On the other hand, robots.txt files enumerate a number of private directory paths. As an example, the robots.txt file of the White House presents hints about sensitive directories, as illustrated in Figure 4.6.

Not only directories but also private documents and images can be searched for with Google.

```
"This document is private | confidential | secret" ext:doc |  
ext:pdf | ext:xls  
intitle:"index of" "jpg | png | bmp" inurl:personal | inurl:private
```

#### *Online Webcams*

Online web cameras come along with their software for remote management over the Internet. Based on the types of webcams, you can filter the url and the title as listed in [25] and access to the online webcam devices without any access control. As an example;

```
intitle:"Live View / - AXIS" | inurl:view/view.shtml
```

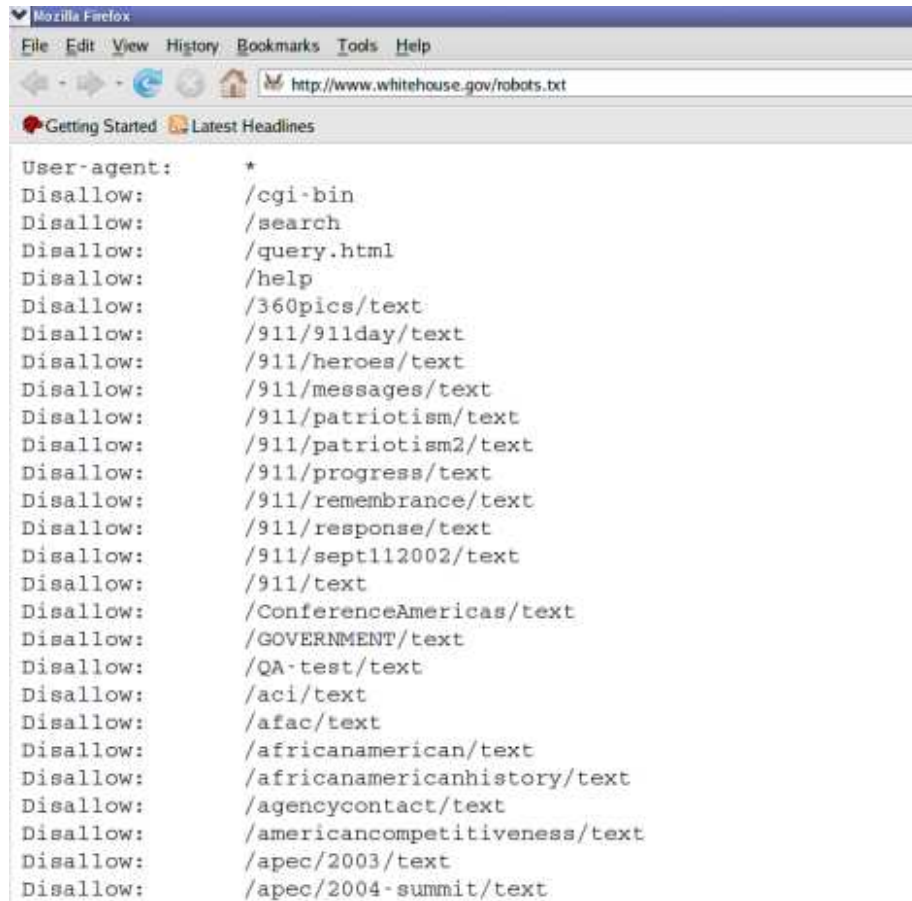


Figure 4.6: The robots.txt file of www.whitehouse.gov

## Secret Data

Secret keys, private keys and encrypted messages comprise secret data, which is expected to be accessible *only* to its owner. The invasions of privacy regarding this group are explained in the following section in detail.

### 4.4.3 Attempts to obtain Cryptographic Secrets

Google can be also used to reveal cryptographic secrets by using its advanced search parameters [184]. Cryptographic secrets searches are grouped into six different groups: *hashed passwords*, *secret keys*, *public keys*, *private*

*keys*, *encrypted messages* and *signed messages*. Public keys are not secret information but included in our queries for the sake of completeness.

### Hashed Passwords

Database structures and contents can be backed up in *dump* files. The following query searches for SQL clauses that may contain usernames and passwords in clear text or in hashed values within dump files. Hash and encryption relevant keywords can also be searched for within files.

```
" create table" "insert into" "pass|passwd|password" (ext:sql  
| ext:dump | ext:dmp)  
  
intext:" password|pass|passwd" intext:"md5|sha1|crypt" (ext:sql  
| ext:dump | ext:dmp)
```

### Secret Keys

Since secret keys are generated mostly as session keys and destroyed after the session is closed, they are not stored on disks permanently. There are, however, still some applications that need to store secret keys, e.g., Kerberos [33] shares a secret key with each registered principal for authentication purposes.

The following query lists the configuration files of a key distribution center (KDC) in Kerberos. Within the configuration files, the path of principal databases which contain principal ids and their secret keys is specified.

```
inurl:"kdc.conf" ext:conf
```

In order to find dumped Kerberos principal databases, the following query can be used:

```
inurl:"slave_datatrans" OR inurl:"from_master"
```

Java provides a tool named *keytool* to create and manage secret keys in key stores. The extension of such keystores is *ks*. The following query searches for java key stores that may contain secret keys. Note that *keytool* can also manage private keys and certificate chains.

```
keystore ext:ks
```

## Public Keys

Public keys, as the name implies, are public information and not secret. For the sake of completeness, the search queries that list public keys are also detailed here.

To list PGP public key files:

```
" BEGIN PGP PUBLIC KEY BLOCK" (ext:txt | ext:asc | ext:key)
```

To list public keys in certificate files:

```
" Certificate:Data:Version" "BEGIN CERTIFICATE" (ext:crt |  
ext:asc | ext:txt)
```

## Private Keys

Private keys should be kept *secret* for personal use but the following search queries show that people do take insufficient care about this and frequently make them publicly accessible.

```
" BEGIN (DSA|RSA)" ext:key
```

```
" BEGIN PGP PRIVATE KEY BLOCK" inurl:txt|asc
```

Gnupg [63] encodes the private key in *secring.gpg*. The following search reveals *secring.gpg* files:

```
" index of" "secring.gpg"
```

## Encrypted Files

For confidentiality, cryptography provides encryption of data. By encrypting, one can store sensitive files and emails securely on local storage devices. The following queries search for encrypted files and emails. It is certain that you need to know the relevant keys for decryption but as shown in the previous examples, it is also possible to find secret keys and private keys. In addition, various cryptanalysis techniques can help to decrypt the encrypted files [171].

The files that are encrypted with GnuPG get the extension *pgp* for binary encoding and the extension *asc* for ASCII encoding. The following first query searches files with *pgp* extension and tries to eliminate signed and public key files from the results. The second query lists ASCII encoded encrypted files. It is the case that signed files have also the same pattern and can be returned with the second query:

```
- " public|pubring|pubkey|signature|pgp|and|or|release"  ext:pgp

- " BEGIN PGP MESSAGE"  ext:asc
```

Many encryption applications use the extension *enc* for the encrypted files. There are some exceptions like AxCrypt File Encryption Software [6] which uses the extension *axx* for encrypted files:

```
-intext:"and"  (ext:enc | ext:axx)
```

In XML Security, the encrypted parts of messages are encoded under *CipherValue* element:

```
"ciphervalue"  ext:xml
```

### Signed Messages

Digital signatures provide integrity, authenticity and non-repudiation in cryptography. The following searches list some signed messages, signed emails and file signatures.

To list *pgp* signed messages (*emails excluded*):

```
"BEGIN PGP SIGNED MESSAGE"  -"From"  (ext:txt | ext:asc | ext:xml)
```

To list signed emails:

```
"BEGIN PGP SIGNED MESSAGE"  "From"  "Date"  "Subject"  (ext:eml
| ext:txt | ext:asc)
```

To list file signatures:

```
- "and|or" "BEGIN PGP SIGNATURE" ext:asc
```

#### 4.4.4 Countermeasures

Google hacking can be very harmful against user privacy and therefore the required security countermeasures should be taken. The protection methods can be grouped as *user-self protection* and *system-wide protection*.

As its name implies, user-self protection requires users to safeguard themselves against the possible threats. Possible countermeasures are as follows:

- Do not make any sensitive data like documents containing your address, phone numbers, backup directories and files, secret data like passwords, private emails, etc. online accessible to public.
- Provide only the minimum required amount of personal information for the Wiki-similar management systems.
- Instead of using a single username over the Internet, try to have more pseudonyms which make linkability of user actions through a single username more difficult.
- Considering forum postings and group mails, try to stay anonymous for certain email contents. Do not mention any company or organization name inside your postings if not required.
- Do not let private media be shared over social networking and blogging services.

As an administrator, you should focus on system-wide protection for the privacy of your users as well. The first method you can enforce is using automatic scan tools [26, 79, 137, 133] that search possible Google threats and test privacy risks within your system. The tools mostly use the hack database [25] when they do the scans. Another method is integration of robots.txt (robots exclusion standard) [54] files into your system. Web crawlers (*hopefully*) respect the directives specified in robots.txt. If this is provided, you can prevent the crawlers from indexing your sensitive files and directories. In addition to this method, you should never make database backups that contain usernames and passwords accessible to public over your system. The most advanced but also complicated method is installing and managing Google honeypots [82] in your system and trying to figure out the behavior of attackers before they attack your *real* system.



#### 4.4.5 TrackingDog: A Penetration Testing Tool for Privacy

To help users to protect their privacy, new privacy enhancing tools are needed. For example, users can be equipped with a penetration testing tool that would search automatically for the possible privacy threats, report its results and warn users. If this is provided, users can be aware of the privacy risks that threaten them.

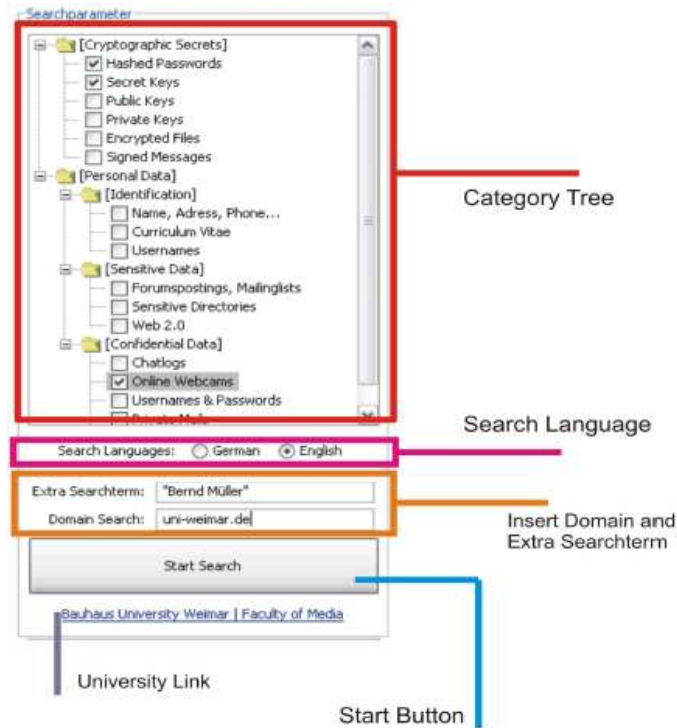


Figure 4.7: TrackingDog - Main GUI

Martin Keßler has implemented the tool namely *TrackingDog* [137] which searches Google exploits mainly for personal data and cryptographic secrets for a given person and/or a given host. *TrackingDog* helps individuals to detect if any of their confidential data have become public over the Internet via Google. It supports both English and German language-specific queries and enables users to edit raw search queries.

Figure 4.7 and 4.8 illustrate the main and result GUI of TrackingDog respectively. In the main GUI, you can choose the queries from the category

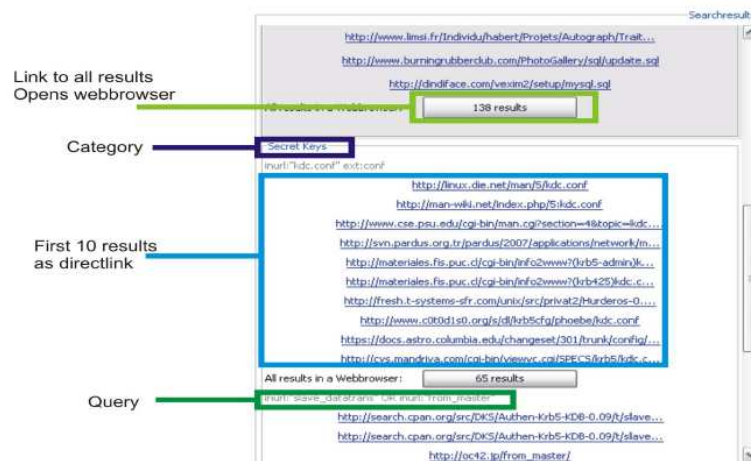


Figure 4.8: TrackingDog - Result GUI

tree and also choose the language for the search. Currently, only German and English searches are supported. You can also enter certain personal names and/or certain host names for your search.

In the result GUI, you get the web links which are found after the search and categorized accordingly. You can click on any result url and open it in a separate web browser.

#### 4.4.6 Related Tools

In addition to TrackingDog, other tools exist which automate the process of security checking via Google. The tools mostly use Johnny Long's database [25]. These database entries are also called as Googledorks.

Gooscan [133] is a Unix/Linux script written by Johnny Long. Based on his database entries, the tool searches for certain key words within a given search engine. For example, executing the following script  
`texttt$ gooscan -t www.google.de -q "index.of picasa.ini" -s de -o output.txt,`  
 the query "index.of picasa.ini" is searched for within google.de (*specified with the parameter -t*). The search is restricted with only web sites from German domains (specified with the parameter -s) and the results are written into an output.txt file. The main aim of this application is helping security experts and administrators to check the security of their systems.

SiteDigger [79], a free application from FoundStone Inc. (a division of

McAfee), can search hacking queries for a given host. Unlike Gooscan, you do not have to provide any search query. It can use the Googledork database as well as its own database and search all queries in the database for the given host. Additionally, it provides a helpful GUI for user interaction as shown in Figure 4.9.

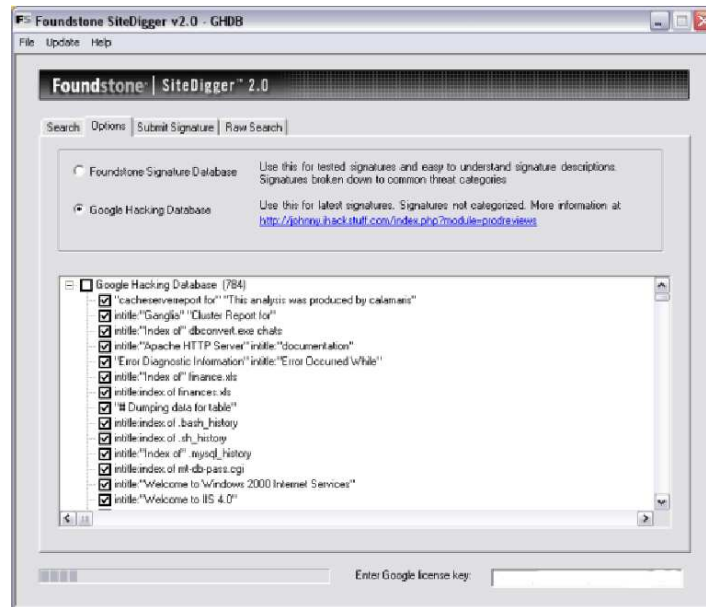


Figure 4.9: SiteDigger - SiteDigger Google Hacking Scanner

Goolink [26] is a very similar tool to Gooscan and uses the Googledork database. AdvancedDork [1] is not a Google hacking tool, but a firefox plug-in which supports searching within Google by using the advanced search key words.

#### 4.4.7 Discussion

Considering the privacy exploits explained in the previous section, one can ask oneself if such exploits are also misused by Google itself to profile people and track their activities. Even though Google replies to this question with a *no* and claims to respect our privacy, most people can not be sure about this dilemma.

In contrast some good approaches to privacy by Google exist as well. Recently, they have declared that they would take steps to further improve

privacy. By searching in Google, your query, your IP and cookie details are stored on the Google servers and that information can identify you uniquely. However, Google has decided firstly to anonymize this collected data within a 18-24 month period [60]. In November 2007, they re-established a new privacy policy and decided to anonymize search logs after 18 months. On the other hand, you can apply other means to remove your cookies from Google servers as explained in [80].

We believe, Google can do more for our privacy. The privacy exploits mentioned above could be taken into consideration by Google. Personal data should not be collected by Google. Internet users are careless and easily make their personal data public unintentionally. This should not be misused by Google. While we hope to gain more respect for our privacy from Google, we also need to help users to get equipped with powerful user-centric privacy enhancing tools such as TrackingDog so as to get to know the threats and to protect themselves.

It is very clear from Google hacking that service providers collecting personal data can directly abuse the privacy of users or cause others to abuse it indirectly by selling or forwarding data. The DAK privacy violation [12] is a good example of privacy abuse. Similarly, service providers of context-aware applications have also the possibility to threaten user privacy. The individuals should also be aware of possible risks and try to prevent misuse of their personal data.

## Chapter 5

# User-centric Proposed Solutions

### 5.1 The SALSA Client Security Architecture

The SALSA client security architecture consists of three main security components. As shown in Figure 5.1, these components are security manager, anonymity manager and storage manager. There are also other relevant security components and they are extending the functionalities of these three main components.

- **Security Manager:** This is the central component in the security architecture. It controls the interactions among other components. It accomplishes the following tasks within the architecture:
  - Management of secure communication with the broker and service providers
  - Management of client authentication over the broker and service providers, i.e. pseudonym and credential managements
  - Management of identity, i.e. personal data such as location, name, address, etc.
  - Management of different security levels for applications (i.e. dynamic security)
- **Anonymity Manager:** This is the main component for supporting anonymous communication based on Mix-nets. It interacts with other components (e.g. Identity Manager) to accomplish its task.

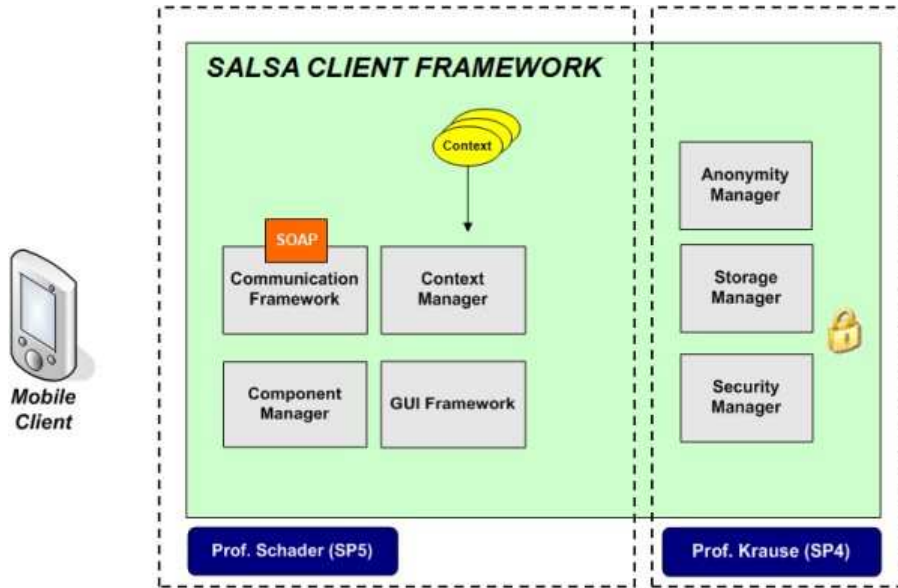


Figure 5.1: The SALSA Client Security Architecture

- **Storage Manager:** This component is responsible for storing confidential data (e.g. personal data, credentials, privacy preferences, policies, etc.) in a secure way (i.e. encrypted) on mobile devices. When other components need to save or restore confidential data, they interact with the storage manager.

The extended security architecture is shown in Figure 5.2. The security manager communicates with the anonymity manager, identity manager and storage manager. The identity manager interacts with the storage manager and the policy manager. The anonymity manager requires interacting with the policy manager in order to retrieve the relevant anonymity policies and enforce the required mechanisms to build anonymous channels. The crypto manager is a complementary component for the storage manager. The security library is accessible to all components in the security architecture. The components of the extended architecture have the following responsibilities:

- **Identity Manager:** Mobile Identity Management enables mobile users to control privacy of their personal data (see Section 5.3). The identity manager supports the functionality of personal data management. Before personal data is sent to any other principal, the identity manager takes this request and accepts or rejects this request based on

user preferences and context. It can even blur the requested personal data before releasing it. For example, instead of sending exact GPS coordinates, the city name can be forwarded or similarly, salary information can be sent in ranges. Mobile users can specify their privacy preferences through the interaction with the policy manager.

- **Policy Manager:** This component is responsible for managing anonymity policies and user privacy preferences. Both policies and preferences are stored in encrypted form thanks to the storage manager.
- **Crypto Manager:** For the encryption or decryption of data, the storage manager needs to obtain the support of the crypto manager which implements certain cryptographic operations based on the security library on the platform.
- **Security Library:** The security library contains different methods for cryptographic operations (i.e. encryption, hash, SSL-based communication, digital signature creation and verification, etc.). In addition to pure Java security APIs, cryptlib [10] security APIs which run faster than Java security APIs on mobile devices have been additionally integrated within the security library of the SALSA client security architecture.

Communicating over untrusted networks (i.e. the Internet), it is a necessity to communicate over secure channels. In order to build secure channels for the communications of mobile users-the broker and mobile users-service providers, SSL-based communications has been implemented as seen in Figure 5.3. The security manager on mobile devices is responsible for creating the secure channels and executing the relevant cryptographic operations. It is not possible to integrate standard SSL Java APIs from the provider Sun onto mobile devices. Hence, Bouncy Castle lightweight Crypto APIs [100] has been used for the implementation.

## 5.2 Dynamic Anonymity

Hiding their real identity is an imperative for people who do not like to share their personal information or secrets with others. They do not want others to get to know their meeting schedules with their business partners, which books they buy and read, how much money they have in their bank accounts, which transactions they execute with their credit cards, etc. Most

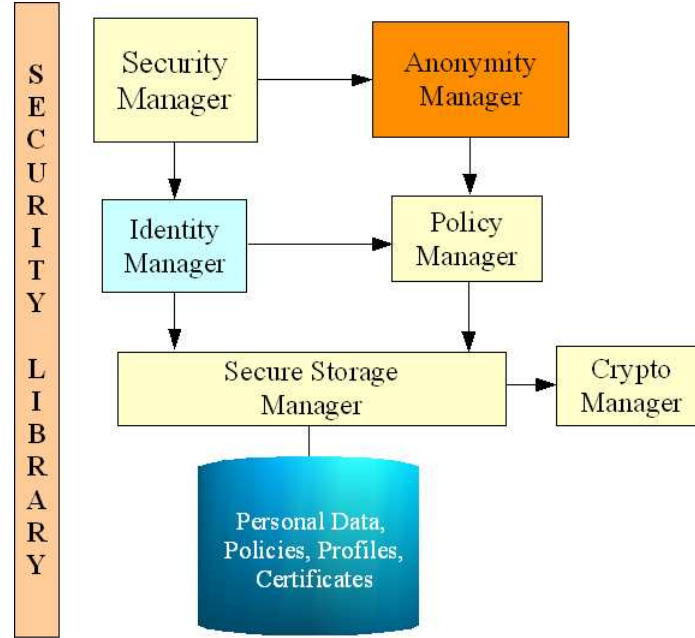


Figure 5.2: The SALSA Client Security Architecture (extended)

people have a strong preference for staying anonymous as far as and whenever possible. In the Internet age, the number of interactions people have with their environment (i.e. their business partners, companies, public service organizations, etc.) has increased enormously. At the same time, the anonymity requirement remains, although it has become more difficult to meet. Staying anonymous when sending e-mails, visiting web pages, and doing e-commerce is needed but not easy to achieve.

Analysis of the security requirements of mobile business environments indicates that anonymity and protection of the mobile user's personal data is one of the greatest challenges [189, 105]. Although security and anonymity are non-functional properties of a system, its user acceptance depends directly on these features.

We distinguish between two types of anonymity – *content anonymity* and *communication anonymity* – which both have to be fulfilled in order to provide complete anonymity. Pseudonyms can provide content anonymity (unobservability [150]) by keeping the users' real identities secret, but an attacker who is able to sniff incoming and outgoing messages on the network nodes can at least find out which nodes are communicating with each other.



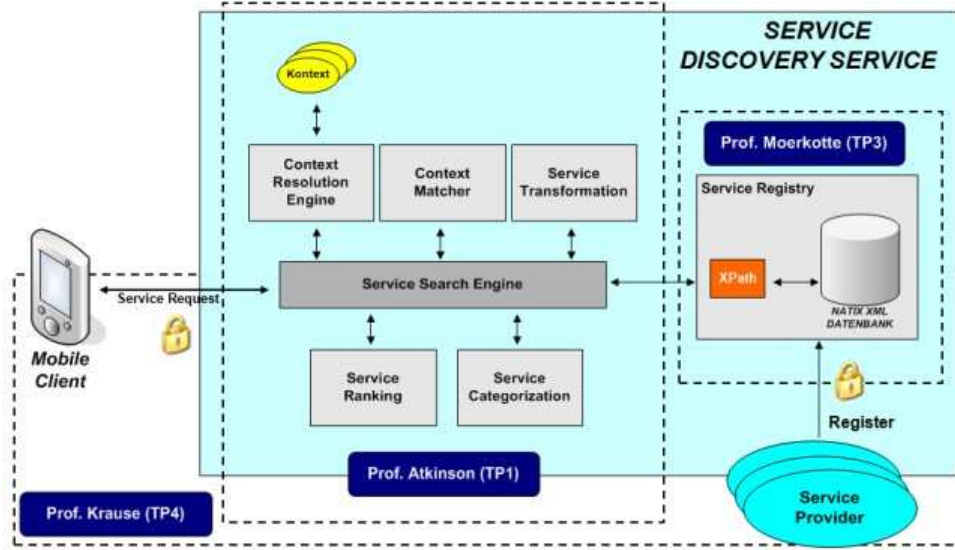


Figure 5.3: The SALSA Client Security Architecture - Secure Communication

This type of threat to anonymity can be averted by ensuring “unlinkability” of user actions [150].

In this section, the solution, namely “Dynamic Anonymity” (that is relevant to communication anonymity and unlinkability of user actions) is explained. Communication anonymity is referred to simply as anonymity in the remainder of the section.

### 5.2.1 Existing Solutions for Anonymity

In the literature, existing solutions for communication anonymity and unlinkability of user actions are categorized into three groups: *proxies*, *peer-to-peer (P2P) networks* and *Mix-net* [129].

In a proxy-based solution, a trusted proxy (anonymizer) receives user requests, rewrites some parts of the request in order to hide sender-specific data and sends it to the final receiver. Replies from the receiver are in turn forwarded to the real sender. The drawback of this scheme is that users have to trust the proxy and there are no protection mechanisms in the channel between users and proxies. For example, [www.anonymizer.com](http://www.anonymizer.com) is a well-known proxy for anonymous web surfing.

With the increasing popularity of peer-to-peer applications, communication anonymity solutions based on P2P networks have been designed (e.g. Dining Cryptographers [102], mCrowds [163], Tarzan [112], etc.). Unlike anonymizer proxies, there is no need for a trusted party within these systems and each user shares an encrypted secure channel with other users in the P2P network. In 1988, David Chaum proposed the “Dining Cryptographer Protocol” for P2P anonymous communication. In this protocol, three or more nodes are arranged over a ring network and each link between the nodes is encrypted. Each participant picks a number randomly and forwards it to the next participant to the right. Each participant then computes the difference between his/her own number and the number he/she received. If a participant wants to send a message, he/she adds it to the difference and announces the result to the others. Then all participants add up the announced numbers. At the end if the sum is 0, that means no one transmitted a message. If the sum is a valid message, that means one participant sent a message. If the sum is an invalid message, that means more than one participant sent a message. Therefore, the protocol needs to be repeated after waiting a random time. In mobile Internet communication, mCrowds [88] presents an anonymity solution for P2P networks. In this solution, the user chooses a random path (a user group) and sends the message along this path to the final receiver.

The last type of solution is Mix-net, which is a more promising approach for the M-Business framework compared to proxies and P2P networks. It was first suggested by David Chaum for anonymous e-mail communication [103]. A *mix* is a computer which resides between a sender and a receiver. When a mix gets a message, it decrypts it and forwards the remaining part to the next mix or the final receiver. A group of mixes composes a network called Mix-net. Chaum’s traditional Mix-net was based on public key operations, but today, Mix-net based solutions relying on symmetric encryption also exist.

In mix-network based solutions the messages are encrypted and exchanged between different nodes positioned between the sender and the receiver. Each node knows about the sender and the next node. Therefore, the first node knows about the user but not the server and similarly the last node knows about the server but not the user.

Mix-net based solutions are well accepted in academia and have also been designed and deployed for different application scenarios. For example, there are solutions for anonymous communication over ISDN networks (*ISDN-Mixes* [151]) and for anonymous email communication (*smtp-remailers* [145]). Jap [32] and Tor [106] are recent Mix-net implementations

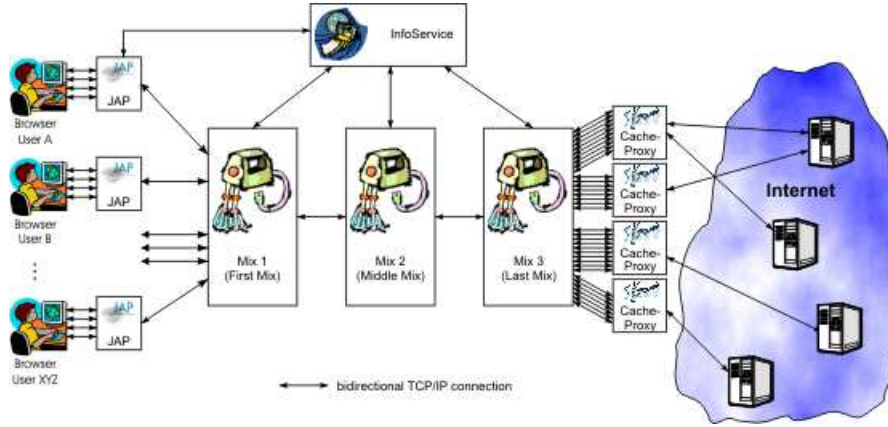
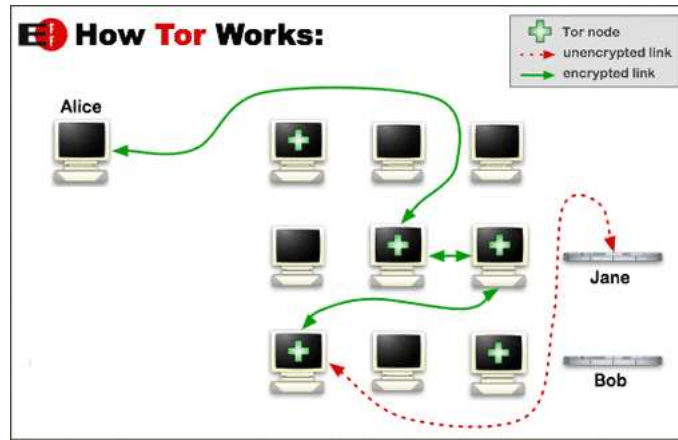
focusing on anonymous web surfing. Their anonymity service is based on the SOCKS protocol [132] and supports any application layer protocol (e.g. ftp, p2p, http, https, etc.). However, Jap does not support protocols other than HTTP due to some organizational reasons, and functions as an HTTP proxy.

Jap has a cascade-style Mix-net (see Figure 5.4), whereas Tor supports free nodes (see Figure 5.5). A Jap cascade consists of two or three mix nodes in a fixed sequential order. The user can only choose the cascade but no particular mix nodes. Supporting free nodes, Tor allows the choice of arbitrary paths randomly through mix nodes.

In Jap, users start the client application and choose a cascade satisfying their anonymity requirement. Each cascade offers different levels of anonymity based on the number of active users on the chosen cascade and traffic parameters. Jap provides both premium and free anonymity services. Premium services enable higher anonymity for paid users. As an example, the free Jap cascade Dresden-Dresden managed by the Jap developers has around 25 simultaneously active users on average [59].

In Tor, since the message route among free mix nodes is chosen randomly, a higher level of anonymity can be achieved compared to the fixed mix order of cascades. On the other hand, since anyone can participate as a mix node within the Tor network, there are certain problems [95, 52]. The Tor designers assume that the traffic between the user and the server is already encrypted. Otherwise, the exit node in the Tor mix-network can sniff the message networks. Based on this restriction, the Swede Dan Egerstad, who was equipped with 5 Tor exit nodes, could sniff around 100 log-in credentials belonging to different consulates in different countries. Jap has also certain advantages over Tor. It supports dummy messages and time delays for higher anonymity levels.

For more details about anonymity, you can refer to the selected papers section of Free Haven (<http://www.freehaven.net/anonbib/topic.html>).

Figure 5.4: Jap Architecture<sup>1</sup>Figure 5.5: Tor Architecture<sup>2</sup>

### 5.2.2 New Anonymity Challenges

Mix-net solutions for anonymity require extra computations. In contrast mobile devices used in context-aware applications are limited in terms of hardware. Therefore, it is not a good idea to integrate Jap or Tor clients into the M-Business framework and thus enable mobile users to communicate anonymously via Jap or Tor networks. Additionally, mobile users and applications may require different levels of anonymity. Applying a fixed level of anonymity can yield weak security or poor performance problems.

<sup>1</sup>Referenced from <https://www.jondos.de/en/>

<sup>2</sup>Referenced from <http://www.torproject.org/overview.html.en>

Operation	Time Consumption on	
	Zaurus SL-C3000 (416 MHz)	IBM Thinkpad R51 (1.7 GHz)
RSA Key Generation ( <i>1024-bit key</i> )	122 seconds	2.2 seconds
RSA Encryption ( <i>1024-bit key, 64-byte data</i> )	172 ms	10 ms
RSA Decryption ( <i>1024-bit key, 128-byte data</i> )	856 ms	40 ms
RSA Signing ( <i>1024-bit key, 64-bytes data</i> )	833 ms	55 ms
RSA Verification ( <i>1024-bit key, 128-bytes data</i> )	169 ms	5 ms
AES Encryption/Decryption ( <i>128-bit key, 2048-byte data</i> )	583 ms	35 ms
SHA-1 Hash ( <i>2048-byte data</i> )	111 ms	5 ms

Table 5.1: Performance of Cryptographic Operations

### Limited Hardware Capabilities

In today's Mix-networks, the sender is required to encrypt a message with the symmetric key of each mix in the message route before sending. Therefore, a key handshake process should be executed between the sender and each mix in the route. These encryption and key handshake processes make a heavy demand on processing power and are consequently time consuming operations that mobile PDAs cannot tolerate. To illustrate, Table 5.1 shows the performance of the required cryptographic operations<sup>3</sup>. The tests were done on both a Zaurus SL-C3000 PDA (416 MHz CPU/64 MB RAM) and an IBM Thinkpad R51 notebook (1.7 GHz CPU/1 GB RAM). For the implementation, we used Bouncy Castle lightweight cryptographic APIs [100]. Note that 100% of CPU power was used during the test computations. Thus, even if the priority of the cryptographic operations is decreased, other applications running on PDAs will hardly receive any CPU power while Mix-net clients are executed.

---

<sup>3</sup>Not all operations in Table 5.1 are required for Mix-net clients, but for the sake of completeness, we have also included the execution times for RSA key generation (normally performed offline), decryption and digital signing.

### Dynamic Anonymity

Both Jap and Tor provide a fixed level of anonymity. You start your client application with a particular configuration, and you cannot easily change or manage your anonymity level afterwards. However, the M-Business framework requires an anonymity feature that enables dynamic updates of anonymity levels for the following reasons:

- *Varying sensitivity of applications:* In the M-Business framework, the anonymity requirements of applications may differ totally. Consider *finding the nearest restaurant* and *mobile dating* applications. *Finding nearest restaurant* is a typical context-aware application. Holding your PDA, you are interested in getting a list of restaurants which are near to your current location. The second application type, *context-aware mobile dating*, lets users search for suitable chat partners within a particular area. Comparing these two types of applications, the latter requires (at least initially) a very high level of anonymity, while the former may not even need anonymity at all.
- *Varying sensitivity of users:* Users also tend to have different sensitivity levels for anonymity. Consider a celebrity interested in having a very high level of anonymity. He/She can even require a high level of anonymity for the *finding the nearest restaurant* application and never wants other people to know the places that he/she eats at.
- *Enhancing performance:* The previous two requirements point out that enforcing a fixed level of anonymity is a security risk since the anonymity level may be too low. An unnecessarily high anonymity level makes, however, applications waste time waiting for cryptographic operations and data transmission delays.

#### 5.2.3 Towards A Solution

Considering varying sensitivity of applications and users and performance problems, we propose a policy-based solution for communication anonymity. The solution emerges existing mix-net based solutions. We have analyzed different Mix-net architectures and found 6 parameters that affect the anonymity level. These parameters can be dynamically specified for each user and application individually within policies and each relevant policy can be enforced when a specific business logic is executed.

### Anonymity Parameters

6 anonymity parameters exist, namely *mix number*, *user number*, *message size*, *message delay*, *time delay* and *dummy message*. Their effects on anonymity are as follows:

1. *Mix number*: This parameter specifies the minimum number of nodes that participate in the Mix anonymization network. If there are more nodes in the message route, traffic analysis becomes more difficult and the anonymity level is increased. Setting this parameter to zero results in a direct client-server communication, i.e. communication without any Mix-net nodes. In the Jap architecture, each cascade consists of two or three nodes. In contrast in the Tor architecture, anyone can join to the Mix-network and become a node. This increases on one hand the anonymity level, on the other hand it can threaten anonymity because there is no control before a node joins into the network.
2. *User number*: In a cascade-style Mix-net like Jap, the number of active users using a particular cascade affects the anonymity level. This parameter defines the minimum number of users that should communicate over a cascade. The more users participate in the communication, the higher is the anonymity level of this cascade.
3. *Fixed message size*: Linkability of messages can be revealed by comparing message size sent over a channel. Preventing this threat, messages should have a *fixed* size. If this is provided, unrelated messages would have also the same sizes, consequently the anonymity set becomes bigger and unlinkability and anonymity are provided.
4. *Message delay*: Upon receiving a message, a mix can either forward the message to the next mix immediately or keep it in its outgoing pool for a certain time period. If messages are immediately forwarded, time delays of messages between mixes can be analyzed for linkability similar to varying message size. Message delay parameter prevents direct sending of messages and specifies the number of messages that should exist in the outgoing pool of a mix before it starts forwarding messages. When the number is exhausted, the mix chooses messages randomly and forwards them to the next receiver. Enabling this option, traffic analysis becomes more difficult, and also additional delays and latencies may be created.

5. *Time delay*: This parameter is similar to *message delay* and prevents immediate forwarding of messages based on time delays rather than message quantities.
6. *Dummy message*: Enabling this parameter, each mix sends extra dummy messages to other mixes when transmitting messages in order to complicate traffic analysis.

#### 5.2.4 The Architecture

Our anonymity architecture focuses on dynamic anonymity and configurable anonymity parameters. In practice there are two approaches for the architecture: *fat* client and *thin* client approaches

##### The Fat Client Architecture

In the fat client architecture, mobile users communicate directly with the Mix networks as illustrated in Figure 5.6. The Mix-net client applications should run on mobile devices and carry out all relevant cryptographic operations (i.e. encryptions and decryptions). This architecture provides strong anonymity, since there is no need for a trusted third party and the first node can only know the mobile user, but not the service provider and similarly the exit node (i.e. the last node) knows the service provider, but not the mobile user. However, this architecture has some bottlenecks in terms of performance and dynamic anonymity parameters.

Not all dynamic anonymity parameters are supported by the existing Mix networks. Even if they are supported, it is not allowed to change these parameters in the client applications. Therefore, a totally new client and server applications should be implemented.

Performance is also a big problem. Based on our preliminary tests as seen in Table 5.1, it takes a very long time for mobile devices to execute primitive cryptographic operations. Setting up a route through the Mix-net would imply sending several messages through a costly wireless connection. Since the initialization of communication routes must be done repeatedly, this would cause a tremendous number of packets to be carried on-air. As a conclusion, even though this approach provides strong anonymity, it has been realized that the computational power of PDAs in the market today is too low to accomplish the fat client approach. Security as a non-functional property of the M-Business framework should not prevent usability.



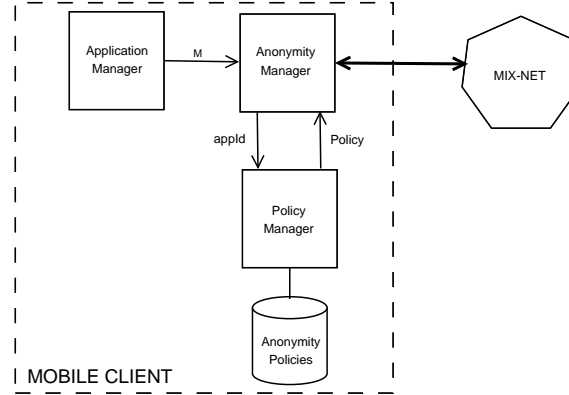


Figure 5.6: The Fat Client Anonymity Architecture

### The Thin Client Architecture

The alternative approach to the fat client architecture is the thin client architecture. As illustrated in Figure 5.7, mobile users connect to a gateway for communicating over the Mix networks. The gateway is a *trusted third party* in the architecture and performs the major part of the computations required for anonymity. This prevents the performance bottleneck due to the Mix networks and it becomes even possible to eliminate latencies only by modifying relevant policies. The gateway residing between mobile devices and the Mix networks acts as a facade [113] and runs different types of Mix-net client applications to forward messages over any client, e.g. Jap-client, Tor-client, etc. In contrast mobile users can still specify the anonymity level for each application individually.

One can argue that this architecture provides the same level of anonymity as a simple proxy architecture. This is not totally true. As in proxy-based solutions, the gateway is also a single point of failure and must be well protected. It is the case that an attacker who *cannot* gain access to a proxy but *can* sniff its outgoing network channels can still break anonymity of users. This scenario is not possible with the thin client architecture, because the gateway does not communicate directly with service providers and therefore does not release the final receiver (i.e. service provider) in message headers to the channel.

Even though we believe the fat client approach is more realistic and provides better anonymity since it needs no trusted party, performance issues on mobile devices did not allow us to implement it for the demo application of

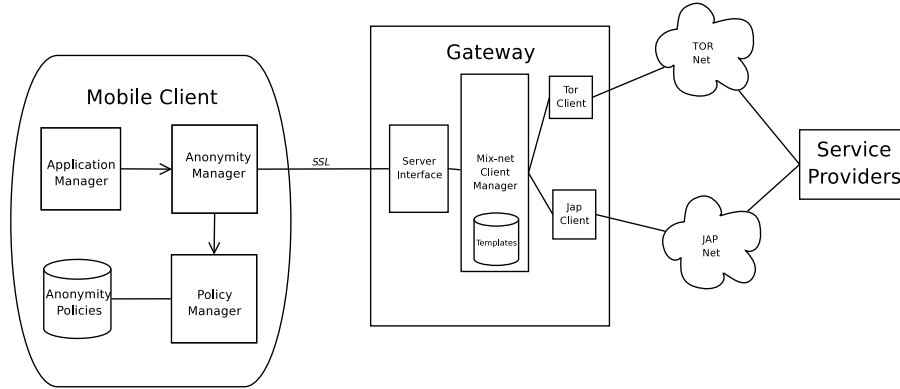


Figure 5.7: The Thin Client Anonymity Architecture

the M-Business research group. By implementing the thin client approach, the performance of anonymity is not a problem anymore, the parameters for dynamic anonymity can be enforced by the gateway and exploiting the existing Jap and Tor networks becomes possible.

*Application Manager*, *Policy Manager* and *Anonymity Manager* are three main components in the architecture on the client-side. Mobile users can create policies for each individual application and specify the required parameters affecting anonymity level. They are stored on mobile devices within the repository of the policy manager. When an application tries to send a message, this request is taken by application manager and then forwarded to the anonymity manager. Afterwards, the anonymity manager asks the policy manager for the relevant policy of this particular application. Based on the parameters in the policy, the application manager decides how to proceed with the message transmission.

The gateway consists of three layers. In the first layer, the *server interface* listens for incoming requests, parses the relevant *destination* and *anonymity parameters*, and forwards them to the second layer, i.e., to the *Mix-net Client Manager (MCM)*. In the third layer, different Mix-net clients are installed and deployed.

The MCM is responsible for managing different Mix-net client applications. The idea behind installing and managing more than one Mix-net client application is that dynamic anonymity can be distributed throughout several network types in order to achieve higher levels of anonymity. Additionally, the MCM can run different instances of the *same* Mix-net client with *different* configuration parameters and thereby enhance the dy-

dynamic anonymity property. On receiving the payload data, destination and anonymity parameters, the MCM chooses the suitable Mix-net client to send the message.

For secure communication, mobile users need to establish only *two* channels, i.e. a channel between mobile user and the gateway and another channel between mobile user and service provider. It should be noticed that enabling secure communications over the Mix networks does not mean that there is no need for secure channels between mobile users and service providers [52].

Christian Beil implemented both the gateway and mobile client applications of this architecture and tested the client software on a PDA. The gateway has been encoded in J2SE and the mobile client in J2ME using a CDC configuration [203]. The gateway has been installed on the Broker-side of the M-Business framework. The performance of our implementation is sufficient for not decreasing the usability of applications while providing a sufficient level of anonymity. For more details of the implementation such as particular design choices and the communication protocol between mobile users and the gateway, you can refer to [96].

### Policies and Templates

Policies and templates exist in the anonymity architecture for the integration of dynamic anonymity features. Each template stored on the gateway specifies a set of certain configurations (i.e. the Mix-net client instance, time delay, message delay and dummy message). Each policy bound to a certain application refers to a particular template contains three additional parameters (i.e. mix number, user number, message size, message padding).

In order to enforce dynamic anonymity policies, the parameters should be supported by the Mix networks. Both Jap and Tor support some of these parameters. In a Jap network, one can choose a cascade based on active user numbers. In the Tor-network, the minimum and maximum node numbers can be configured. Tor also supports creating a new circuit after sending a particular number of messages over an established channel. The *dummy message* and *time delay* parameters are supported by Jap but it cannot be changed on the client side. Therefore, the gateway emulates their effects. It creates dummy Mix-net clients, and if a value is set for the dummy message parameter, dummy messages are sent over this dummy Mix-net client. If the time delay is set, the gateway keeps the message for a particular time and then sends it over the Mix network. Applying time delays, timing attacks against exchanged messages are prevented. Additionally, as an enhancement

to dynamic reconfigurability, mobile users can specify within policies which Mix-net client to use.

```
<anonConfig>
  <defaultAnonTemplate id ="default">
    <general>
      <parameter name="dummyMessages">0</parameter>
      <parameter name="timeDelay">0</parameter>
    </general>
    <provider name="Tor" host="localhost" port="9050" protocol="socks4a" />
  </defaultAnonTemplate>

  <anonTemplates>
    <anonTemplate id="jap1">
      <general>
        <parameter name="dummyMessages">5</parameter>
        <parameter name="timeDelay">3</parameter>
      </general>
      <provider name="JAP" host="localhost" port="4001" protocol="socks4a" />
    </anonTemplate>

    <anonTemplate id="tor1">
      <general>
        <parameter name="messageDelay">10</parameter>
      </general>
      <provider name="Tor" host="localhost" port="9050" protocol="socks4a" />
    </anonTemplate>

    <anonTemplate id="direct_withDelay">
      <general>
        <parameter name="timeDelay">2</parameter>
      </general>
    </anonTemplate>

    <anonTemplate id="direct_default" />
  </anonTemplates>
</anonConfig>
```

Figure 5.8: Samples of Anonymity Templates

Figure 5.8 show samples of anonymity templates encoded in XML. Each template file contain a default anonymity template encoded in *defaultAnonTemplate* tag and also other specific templates encoded in *anonTemplates*. For each template, general and provider parameters can be specified. The

provider parameter represents the relevant Mix-net, i.e. Jap, Tor, etc. If there is no provider specified, this results into direct communication with service providers. In the general tag, the parameters, time delay, message delay and dummy messages can be specified. Each template has a unique identification and user policies can refer to template ids.

```
<anonPolicies>
  <policy appID="app1">
    <anonTemplate>tor1</anonTemplate>
    <messageSize>512</messageSize>
    <messagePadding>true</messagePadding>
    <minMixNumber>3</minMixNumber>
    <minUserNumber>500</minUserNumber>
  </policy>
  <policy appID="app2">
    <anonTemplate>jap1</anonTemplate>
    <messageSize>2048</messageSize>
    <messagePadding>false</messagePadding>
  </policy>
  <policy appID="app3">
    <anonTemplate>direct_default</anonTemplate>
    <messageSize>1024</messageSize>
    <messagePadding>false</messagePadding>
  </policy>
  <policy appID="app4">
    <anonTemplate>direct_withDelay</anonTemplate>
    <messageSize>1024</messageSize>
  </policy>
</anonPolicies>
```

Figure 5.9: Examples of Anonymity Policies

Figure 5.9 illustrates sample anonymity policies and their bindings to particular templates. Each policy can be bound to one or more applications and refer to a certain template. In addition, message size, message padding, minimum mix number and minimum user number parameters can be specified in policies. The message size parameter specifies the maximum size of sent messages, and the message padding parameter specifies whether a fixed or variable message size is used.

### 5.2.5 Threat Analysis

During the communication, mobile users connect to the gateway and open a channel through the Mix networks to service providers. Upon receiving a message from a mobile user, the gateway encrypts the message and sends it through the Mix network to the relevant service provider. In the threat analysis, the goal of an attacker is assumed as revealing the relations among communication partners by applying traffic analysis.

In the Mix networks, attackers can be categorized as follows:

- *Passive or Active Attackers:* Passive attackers are able to sniff and capture the messages exchanged over the networks. Based on their facilities, they can seize messages sent from mobile users to the gateway, messages sent among Mix-net nodes and messages exchanged between the exit nodes in the Mix-net and service providers. Active attackers can enforce all the sniffing attacks of passive attackers. Additionally, they can insert, alter, or drop the exchanged messages.
- *External or Internal Attackers:* Attackers can be grouped as external or internal based on their control over the principals (i.e. nodes or the gateway) of the Mix networks. Controlling one or more mix nodes, internal attackers can distinguish between dummy messages and actual data messages and can link incoming and outgoing messages of their controlled nodes. Controlling the gateway is unacceptable for the thin client architecture, since the gateway knows the sender and the receiver of any message. An external attacker has no control over any node or the gateway and therefore is less harmful compared to internal attackers.
- *Partial or Global Attackers:* An attacker is “partial” if he or she can influence only a part of the system. A global attacker can, however, potentially threaten the entire architecture.

An attacker is commonly assumed to be global, passive, and external. However, since the Mix-nets that underlie our architecture are not secure against such attackers, our system is not, either. Additionally, a compromised gateway would completely expose the identity of all users running their traffic through it. However, users are reasonably protected against active, internal and partial attackers who control only small number of mix nodes, not including the gateway. We expect that attackers monitoring the connection between mobile users and the gateway and controlling some

nodes in the Mix networks and a number of service providers would not compromise our anonymity model.

The intentions of attackers may vary. A service provider may be interested in the identities (i.e. e-mail and postal addresses, etc.) of its competitor's customers in order to target them with specific advertisements. Another possible attacker is users who try to find out the providers with whom their target user is communicating. Similarly, for the Friend Finder applications, attackers can attempt to ascertain the relationships between mobile users.

### 5.2.6 Strengths and Weaknesses of the Architectures

Two architectures (i.e. the fat client and the thin client) were proposed in this section. Both architectures fulfill the requirements of dynamic anonymity, i.e. applying configurable anonymity parameters.

The fat client architecture support a high level of anonymity compared to the thin client architecture. It requires, however, a very large amount of computational power which is lacking in small mobile devices.

In contrast the thin client architecture can support dynamic anonymity and has the computational power available for anonymity promoting cryptographic operations. Another advantage of this architecture is that it promotes weak coupling between mobile devices and particular Mix-net implementations, such that existing and well-established Mix-networks can be integrated in a way that is transparent for mobile devices. Especially when support of an additional Mix-net implementation is to be added to the framework, the client component must only be deployed to the gateway but not to mobile devices. From both security and software engineering points of view, this is a very desirable property.

As a drawback, the thin client architecture requires an extra trusted third party and this is a threat point for anonymity. In case the gateway is compromised, anonymity fails. Therefore, it is very important as to who operates the gateway. In the implementation, the broker, who already acts as a trusted third party from the business point of view, deploys the gateway since it is already a trusted third party in the M-Business framework.

### 5.2.7 Future Work

Supporting a higher level of anonymity, the fat client architecture should be implemented for the M-Business framework if future mobile devices possess

more computational power and can cope with time-consuming cryptographic operations.

The concrete effects of the defined anonymity parameters and their comparison in terms of anonymity are open questions. For example, the concrete effects of time delay and dummy messages regarding anonymity level are still not well defined. Having implemented the thin client architecture, the next step should be the study of the configuration parameters and their effects on anonymity level.

Configuration parameters encoded within policies and templates provide dynamic anonymity. However, it is neither realistic nor practical to expect non-technical mobile users to specify their own policies and parameters for each application. This process should be as easy as possible. For example, users can choose from pre-defined anonymity levels, e.g. ranging from *high* to *low*, for individual applications or the entire M-Business client.

### 5.3 Mobile Identity Management

Social, ethical and legal aspects require the protection of the privacy of users on the digital Internet platform. With the introduction of new web technologies, users transfer much of their personal data to other service providers and Internet users. The danger of misuse of the collected personal data threatens the privacy of users. Service providers can profile users, send spam emails based on their profiling results, apply dynamic pricing which means different people pay different amount for the same service, forward and even sell data collected to third parties.

In contrast the privacy protection laws based on EU directives [15, 16] regulate that personal data of an individual should not be retrieved without his consent, not be used for other purposes rather than the stated purpose, not be shared with others if that is not agreed before and be deleted if the user withdraws his or her consent later. Technical system developers should take the privacy regulations into consideration and integrate privacy-enhancing tools within their systems in order to help to guarantee the privacy of individuals.

Identity is described as “one or more attributes, which are applicable to this particular subject or object”<sup>4</sup>. A user can possess many identities and each identity can be assigned to a number of his differing attributes. Identity management helps individuals to control their personal data when it needs to be shared with other parties and thus supports their privacy.

---

<sup>4</sup>Referenced from Wikipedia: [http://en.wikipedia.org/wiki/Identity\\_management](http://en.wikipedia.org/wiki/Identity_management)



Mobile users of context-aware applications also have identities that are in interaction with other principals. In particular, the *location* attribute of mobile users is a very sensitive context attribute and must be protected against unauthorized access. Considering this, mobile identity management has become an important requirement for context-aware applications.

Mobile identity management can be considered as a subgroup of identity management. This statement is only partially true. The new context data, especially location data, has its own characteristics and therefore not all identity management solutions can be applied to mobile identity management.

In this section, user-centric identity management within the Friend Finder application is focused on and the most important privacy aspects from the perspective of mobile identity management are examined and evaluated. The aspects are not specific to mobile identity management, but their evaluation is mobile-centric.

### 5.3.1 Related Work of Privacy and Identity Management

Privacy and identity management are very active research topics in the academic world. Many research projects exist focusing on different aspects of supporting privacy and identity management.

P3P (Platform for Privacy Preferences) [51] and Appel (A P3P Preference Exchange Language) [3] are W3C recommendations and help individuals to build a trust relation with servers and service providers. Servers and service providers specify their data collection policies as P3P policies and publish them. Users specify their privacy preferences in Appel format. Before users communicate with servers, the P3P-capable user agents (e.g. browsers) retrieve the server's P3P policy on behalf of users, compare them with the users' Appel preferences. If any conflict exists between the policy and the preferences, either the users are asked how to proceed or communication is stopped.

E-P3P (Enterprise Privacy Practices) [89] was IBM's first attempt to propose a privacy model to support privacy requirements and enforce policies within enterprises. EPAL (The Enterprise Privacy Authorization Language) [90], as a successor of E-P3P, was designed by IBM and submitted to W3C to become a standard. EPAL is an XML-based privacy policy specification language aimed at organizations to enable them to formalize internal privacy policies. Unlike P3P, EPAL takes into consideration the enforcement of privacy rules and focuses on the B2B privacy domain.

Myles et al. propose an architecture for preserving privacy in environ-

ments with location-based applications [108]. They extend the P3P policy language in order to cover location-based applications as well. Mobile users initially send their privacy preferences called *validators* to the central location server. To make a location request for a particular user, the service providers need to send their privacy policies to the location server. Afterwards, the validators are evaluated with the request and the relevant privacy policy. If this process is successful, the location is sent to the provider.

The Geopriv project group [81] focuses on location privacy, i.e. authorization, integrity and privacy requirements for the transfer of location information. The group has defined a standard for the secure and authorized transmission of location information (e.g. coordinates, postal addresses, etc.) and privacy policies over the Internet.

Confab [120] provides a customizable framework for building ubiquitous computing applications. Through the analysis of the privacy requirements of end-users and application developers, the framework also provides extensions for managing location privacy and trust levels in applications.

*pawS* [130] is a privacy awareness system for ubiquitous computing environments. It uses P3P policies to specify privacy concerns for collected data and the Appel language to specify the user privacy preferences. The mobile device runs a mobile privacy assistant that communicates with the ubiquitous devices, receives their privacy policies, compares them with the user preferences and finally accepts or rejects the communication with the ubiquitous device.

In the WASP project [74], the Appel language is extended to support context-based applications. The extensions add support for date, time, day of the week and location entities in the preference language. This is a good approach to show that P3P can be extended to support context-based applications. However, these extensions only support basic context data. More context data and the rules of context-to-context relations should also be integrated within the preference language (see Section 5.3.4). Providing this, if some context values do not satisfy certain privacy conditions, sending this context data can only be refused and using the service is continued as well.

The PRIME (*Privacy and Identity Management for Europe*) project [53] is supported by the European Union's Sixth Framework Program and the Swiss Federal Office for Education and Science. The aim of PRIME is the development of privacy-enhancing tools for identity management. *idemix* [29] has been developed within the PRIME project and aims at achieving anonymous *authentication* in applications. The LBS [42] prototype of the PRIME implements a demo of location-based applications for pharmacy search. The demo application shows how privacy of personal data and pseudonymity

can be protected against possible violations by mobile operators and service providers. Unlike our focus on push services, the demo application considers only pull services. In addition, certain aspects like context relations and dependencies, blurring in levels are not within the scope of the PRIME LBS.

The FIDIS (*Future of Identity in the Information Society*) [18] project is a Network of Excellence project and is supported by the European Union under the 6th Framework Programme for Research and Technological Development. FIDIS focuses on topics such as future identity management, identity theft, privacy in a legal-social context, mobility and identity, etc. They have compiled a detailed database of identity management solutions in academia and industry [19].

The NEXUS project at the University of Stuttgart focuses also on context-aware applications and “*envisions the World Wide Space to be the common basis for future context-aware applications*”. In their sub-project regarding security and privacy, they propose [118] providing location privacy by applying coordinate transformations. They show how location can be rendered illegible and yet still facilitate the possibility of performing processing operations required by location-based services.

Jendricke et al. present an identity manager to control personal data sent from mobile devices through networks [122]. An identity manager provides an interface with which one creates different virtual identifications (IDs), i.e. pseudonyms, and binds a subset of his personal data to each ID. When communicating with a service provider, the user chooses an ID that is suitable for this particular type of communication. Before any personal data is sent to a service provider, the user is explicitly asked to confirm the transmission. However, it should be realized that the identity manager covers only limited aspects of mobile identity management for context-aware applications. As examples, blurring in levels, history management, trust management and context relations are not explicitly supported.

### 5.3.2 User-controlled Mobile Identity Management

Mobile users have security considerations and are anxious about the privacy of their context, especially their location [86]. In many cases, they require guaranteed context privacy, otherwise, they would refuse to use the service. Some typical and specific questions regarding context privacy are:

- What happens if service providers collect my location information regularly and use this collected information to track me or to make user profiles?

- What if service providers record at what times during the day I frequently use the services and send me advertisements at these times?
- What happens if the service providers share my location information with third parties?
- Do service providers really need my location for this particular service?
- Is it enough if I give away my location information with a low resolution?

Mobile identity management helps users interacting with mobile applications to safeguard their personal data in the digital world as they do in the physical life. You have many relationships with other people and organizations in society. You are, let's say, a computer scientist, a husband, a child, a friend or a stranger to a range of various people - all at the same time. That means that you have many *partial* identities [150]. Each partial identity is mapped to a group of attributes. You can intuitively decide which partial identity is used for communicating with whom. You also switch from one partial identity to another very easily and quickly. You can control as to whom you trust and do not trust. This scenario also needs to be possible in the digital world. You should be able to create different partial identities, map a group of attributes to this identity and decide which partial identity to use based on your communications partner.

Mobile identity management should go beyond the formulation of "*which attributes belong to which identity and which identity is used with whom*". If you consider normal social life and conventions, you do not give away any information about yourself to someone who might give this information to other people. You build your trust relations automatically (and mostly) unconsciously with others. You do not talk about your secrets with your friends if others are around and can eavesdrop – this is something like *secure* communication. Considering one's *location* attribute, you do not reveal your exact location to everybody endlessly. You often relate your exact location in weekdays to your boss but at the weekend your boss does not need to know your location (indeed, normally has no right to do so). However, it is generally considered reasonable that your wife/husband can get to know your location at anytime – this being independent of where you are. You generally remember which information was given to which person (i.e. history management). If required, you prefer staying anonymous while you are in interaction. When you buy something from supermarket, you do not need

to (or want to) identify yourself. All these aspects are part of identity management and must be supported by mobile identity management solutions in the digital world as well.

### 5.3.3 Privacy Policy in P3P

P3P aims to protect web users against Internet privacy risks. P3P server policies are encoded in machine-readable XML format. Within a P3P policy, a service provider can specify its identity data, the data it collects and the reason, the retention period, the dispute policy, whether the users can be identified with the collected data and the parties that can access the data. In addition, the users specify their privacy preferences in Appel [3]. Before a user starts communication with a service provider, the user's P3P-enabled agent retrieves the provider's P3P policy, compares it with the privacy preferences and interacts with the user in order to decide how to proceed in case there is a conflict between the server's policy and the user's preferences. If there is no conflict, the agent initializes communication with the service provider.

P3P does not cover all user privacy aspects in context-aware applications, because the main consideration of P3P is only the interaction between users and service providers. In contrast, the user's privacy issues are related to the user himself/herself, the environment and other users as well. P3P only controls data collection and forwarding privacy aspects. In Figure 5.10 and 5.11, a part of a sample P3P policy is given to show a typical privacy policy specification.

Data such the name and contact information about the policy holder are specified in the *ENTITY* tag. The policy specifies with the *ACCESS* tag whether the user is allowed to view or update his/her collected data. In this policy, the users are given access to all identified data. The privacy holder defines the possible solutions for any disputes under *DISPUTES-GROUP* tag. For example, this policy specifies that in case of any dispute the customers can contact the customer service department. As a remedy the error can be corrected, also the relevant law can specify the remedies or even monetary damages can be paid to the users.

Each *STATEMENT* tag specifies a group of personal data, the purpose of data collection, the consequences of various actions, identifiability, who can access the data, and the retention time. The *PURPOSE* tag specifies that the collected data can be used for both the main purpose which is for the distribution in the Friend Finder application (implied by *<current/>*

```

<ENTITY>
<DATA-GROUP>
  <DATA ref="#business.name">Location Provider Service</DATA>
  <DATA ref="#business.contact-info.online.email">p3p@example.com</DATA>
  <DATA ref="#business.contact-info.online.uri">http://www.example.com</DATA>
  <DATA ref="#business.contact-info.postal.organization">University</DATA>
  <DATA ref="#business.contact-info.postal.street">University Address</DATA>
  <DATA ref="#business.contact-info.postal.country">Germany</DATA>
</DATA-GROUP>
</ENTITY>

<ACCESS><all/></ACCESS>

<DISPUTES-GROUP>
  <DISPUTES resolution-type="service"
    service="http://www.example.com/p3p_dispute.html"
    short-description="Dispute">
    <LONG-DESCRIPTION>
      For any inconvenience, apply to our Customer Service (dispute@example.com)
    </LONG-DESCRIPTION>
    <REMEDIES><correct/><money/><law/></REMEDIES>
  </DISPUTES>
</DISPUTES-GROUP>

```

Figure 5.10: P3P Sample Policy

tag) or individual analysis for determining individual characteristics (implied by *<individual-analysis/>* tag). RECIPIENT tag has the value *<ours/>* which means only the service provider and its agents can access the personal data. RETENTION tag has the value *<stated-purpose/>* which requires information to be deleted at the earliest time possible.

### Shortcomings of P3P/Appel

A user's privacy preferences can be related to the service provider, the user himself, other users and the environment as depicted in Figure 5.12. P3P considers the privacy concerns of a user *only* in relation to service providers. Therefore, the Friend Finder needs more comprehensive privacy policies for user-centric privacy management.

P3P policies and Appel preferences should be extended to cover all user privacy aspects. The shortcomings of P3P/Appel are relating to all privacy factors are explained in the following subsections:

#### *Factors relevant to Location Provider*

By comparing P3P policies and user preferences, privacy of users can be

```

<STATEMENT>

  <EXTENSION optional="yes">
    <GROUP-INFO name="Location"/>
  </EXTENSION>

  <CONSEQUENCE>
    Location data will be collected with the aim of enabling the service.
  </CONSEQUENCE>

  <PURPOSE>
    <current/><individual-analysis/>
  </PURPOSE>

  <RECIPIENT><ours/></RECIPIENT>

  <RETENTION><stated-purpose/></RETENTION>

  <DATA-GROUP>
    <DATA ref="#dynamic.miscdata"><CATEGORIES><location/></CATEGORIES></DATA>
  </DATA-GROUP>

</STATEMENT>

```

Figure 5.11: P3P Sample Policy (cont.)

controlled. There are, however, still some shortcomings of P3P and Appel as regards the privacy preferences of users.

Policy negotiation is not possible with P3P. However the privacy sensitivity of users can vary and therefore service providers should be in the position of presenting different privacy policies that enable negotiations to be undertaken with differing users and their preferences.

P3P policies are static and do not support dynamic evaluation. For example, quality of service based on the blurring level of the location cannot be expressed in P3P. As a dynamic privacy aspect, a user may want to be navigated to a certain restaurant if he gives his exact location. If he blurs his location and gives only the zip code for example, then he could get a list of the restaurants in that area as plain text. Such dynamic behaviors are not considered in the P3P specification.

*Factors relevant to the User (i.e. location owner)*

At certain conditions, the user may not want to participate in the Friend Finder service and reveal his location to others. As examples:

- at certain dates and times, e.g. on holiday, in the evenings, at the

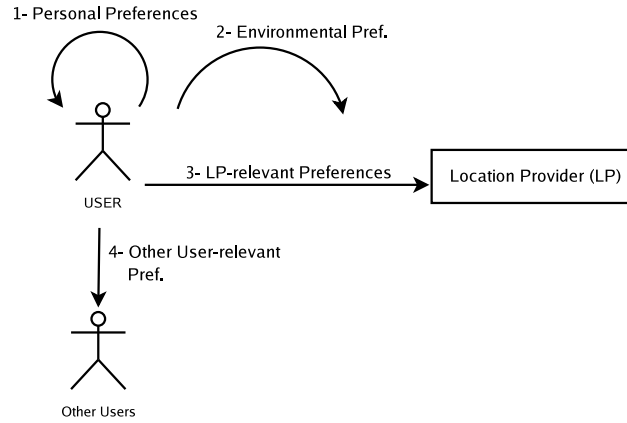


Figure 5.12: Privacy Concerns of Users

weekends, etc.

- based on the mood or status of the user, i.e. if the user is very unhappy or away.
- based on location (if his location is a certain place, e.g. X street, only then he reveals his location).

#### *Factors relevant to Other Users*

Based on the identity and context of other users, the user may not want to reveal his location to particular users. As examples:

- Only the users with certain identities can access the location of the user.
- Only the users that are at a certain location can access the location of the user, e.g. the users that are in the same building as the user is in.
- Only the users that hold certain context data, e.g. users having similar hobbies as the user himself

#### *Environmental Factors*

The user's privacy concerns can also be affected by the environment such as application type (i.e. indoor/outdoor application), physical conditions (e.g. light, pressure, etc.), network infrastructure, etc. As examples:



- The user participates in the service if the service is at an outdoor application.
- In outdoor applications, the user releases his exact location (e.g. GPS coordinates). Otherwise, he wants to blur his location and reveals only building names instead of the floor name and/or room name/number.

### Extensions to P3P/Appel

P3P and Appel need to be extended to enable them to be integrated within an all-embracing privacy architecture for the Friend Finder or generally speaking for context-aware applications. They should be extended in such a way that they support negotiation and dynamic evaluation. Moreover, the policy and preferences languages should be extended to support context-based features stemming from the user, the environment and other users as explained above. The context-based features of the user can be the user identity (e.g. name, address, phone number, etc), the user profile (e.g. user interests, schedule, etc.), his morale, his busyness, location and time. The features of the environment can be physical conditions (e.g. light, pressure, etc.) and the network infrastructure (e.g. indoor application, outdoor application, etc.). The features of the other users are the same features that the user himself has.

A close relationship exists between all these features in terms of privacy. Each feature can affect other features in terms of privacy level. For example, the feature identity is affected by the feature morale status. The user may not want to share his location with another user if he is angry with him. Or similarly, if the user is very unhappy, he would stop sending his location data to the location provider in order not to let others know his location. As an example for the relation between location and the network infrastructure, the user would blur his location for indoor application, whereas he gives his exact location for outdoor applications. It is clear from these examples that any feature can be a privacy evaluation factor for another feature and P3P/Appel should be extended in such a way that all these feature-to-feature relations can be expressed in policies for user privacy management.

In Figure 5.13 depicting the feature relationships, it is shown that to protect the features of the user and the environment, it should be possible to evaluate all other features within privacy policies and preferences. After the evaluation, the original private data can be released to others, or the blurred/falsified version is released or the data is not transmitted.

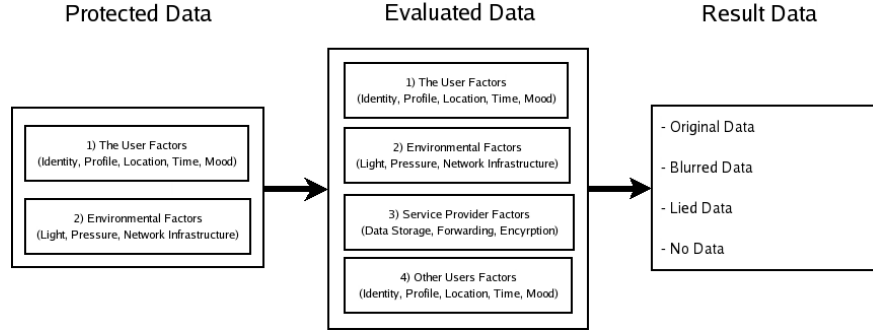


Figure 5.13: Feature Relations for Privacy

### 5.3.4 The Aspects

Based on their experiences and analysis of existing systems, Lederer et al. [131] explain 5 pitfalls possibly met with the design of technical systems related to personal privacy:

- The first pitfall is *obscuring potential information flow*. The technical systems should let users know what kinds of information are being collected about them, its purpose, duration and the receivers.
- The second pitfall is *obscuring actual information flow*. Users should know exactly what actions are executed and nothing should be hidden. For example, if a cookie is set-up on a users' device, the user should be informed.
- The third pitfall is *emphasizing configuration over action*. The systems should not require very many configurations and expect users simply to live with them.
- The forth pitfall is *lacking coarse-grained control*. Users should be in the position of canceling any data transfer or blurring of personal data.
- The last pitfall is *inhibiting established practices*. Designs of technical systems should employ privacy patterns (e.g. blurring, anonymization, data limitation, etc.) effectively.

Considering all these pitfalls, the required aspects for identity management are explained and evaluated from the perspective of the Friend Finder

in this section. The aspects are not directly specific only to mobile identity management, but their evaluation is specific to context-aware mobile applications.

### Context-to-Context Dependence

A user can have both *static* context data such as name, surname, address and *dynamic* context data such as current location, local weather conditions, velocity in his car, etc. There is a very tight dependence between different context data in terms of privacy. For example, a name-surname pair is dependent on address data and vice versa. If someone knows your name and surname, it is not so difficult to find out your address. That means if you give away your name-surname, you also give away your address. Similarly in location-aware applications, if you know someone's current location with the velocity and direction, it is not difficult to reveal his future location within a one hour. Considering the Friend Finder application, it is not difficult to find out the identity of a particular person if you get to know his friends list. Mobile identity management systems should allow users to specify their privacy preferences. You can explicitly specify which of your personal data is released or not released.

The *context2context dependence* aspect should be integrated within mobile identity management system of the Friend Finder application. In case any logical conflicts exist during data release, mobile users should be warned and asked how to proceed by the mobile identity management system. If this is provided, the pitfalls regarding "obscuring potential and actual information flow" can be avoided.

### Context-to-Context Relation

During the management of privacy preferences, your choices are affected by the relations between different context data. As a simple example, there is a relation between the *location* and *time*. You can specify a preference such as "*I do not want to release my location at the weekends*". Similarly, a *location:(time,person)* relationship can also exist. "*I want my boss to get to know my location only at weekdays*" is an example of such a relationship. A *location:own\_location* relationship can be explained with the example "*I do not want to reveal my location if I am in Stuttgart*". The relation *location:remote\_location* can be given as "*I do not want to reveal my location if the other party is not in the same building as me*". Static data can also have this kind of relation. *interest:interest* relation for location-based chat means

that “*I release my interests only to people who hold the same interests*”.

The *context2context* relation aspect should be also a part of the mobile identity management system of the Friend Finder application. Its privacy preference language should support specifying context2context relations in terms of privacy. This aspect, as a privacy pattern, avoids the pitfall of “inhibiting established practices”.

### Blurring in Levels

Blurring of a personal data means revealing personal data not in an exact form but rather in ranges or in a more abstract form. Blurring can help to protect privacy and identity. You can give out your exact salary, but this data can give hints about your job status and life standard. If possible, salary should be given in ranges which make such conclusions more difficult. Location blurring can also be applied. Location tracking can be prevented by applying blurring. For certain applications, it should be enough to give only the city name or zip code<sup>5</sup> instead of exact GPS coordinates. For indoor applications, location blurring can be also very helpful. Giving an exact room number (which you are in, in a hotel) is not something you would prefer to reveal unless really necessary. Instead, you can blur it and reveal only the building name if it does not hurt the functionality of the application.

Blurring can be applied in levels. For example, for outdoor locations; GPS coordinates, street name, zip code, city, country and continent can compose such location levels. For indoor locations; room number, floor number and building name can define the levels. Blurring in levels can also be used to improve the quality of service. Assume you are holding your PDA and are in the Stuttgart city center. You are interested in finding near-by restaurants. You can either give your exact GPS coordinates or you can reveal your zip code (PLZ) or city name. If you release your GPS coordinates, the service provider provides a map graphic which directs you to different restaurants in the neighborhood. If you give only the zip code, then you get the restaurants as a simple text list with addresses.

Blurring can improve the privacy and also the quality of service. The pitfall “lacking coarse-grained control” can be avoided by the integration of blurring in levels aspect. Hence, the mobile identity management system of the Friend Finder application should support blurring mechanisms for any possible context data.

---

<sup>5</sup>It should be noted that zip codes differ between countries – for example, in the UK the Post Code sometimes identifies a unique house/flat.

### Extensible Preference Language

The specification language for privacy preferences is very important for mobile identity management. Appel [3] as a privacy preference language has limitations as to identity management [115]. Selection of a group of personal context data to release based on privacy policies is not fully supported by Appel to be integrated within context-aware applications. Therefore, the preference language for the Friend Finder application should take into consideration explicitly the different static and dynamic context data, their dependencies and relations and also blurring in levels.

The proposed extension for such a preference language is illustrated in Figure 5.14. User privacy preferences are encoded in xml format. The mobile identity management system of the Friend Finder rejects any data release unless any exception has been defined by the mobile users for a particular role (i.e. group of persons) or a person. The context data to be protected is defined with the tag *protected* and its attribute *property*. Each protected element contains one or more *exception* tags which consist of the attributes *role* and *id* and *if* tags for the validation of the exceptions. The exception can be defined for a certain group with the attribute *role* or a certain person with the attribute *id*. An exception is also evaluated as true, if only all *if* conditions are validated as true. Each *if* tag contains a *context* attribute (i.e. the context data for validation), a *condition* attribute (i.e. the comparison structure) and a *value* attribute.

```
<protected property="location|name|interests|velocity">
  <exception role="family|work|private|..." id="wife|boss|..">
    <if context="location" condition="is|is-not" value=" " />
    <if context="time" condition="is|is-not|before|after" value=" " />
    <if context="interest" condition="similar|not-similar" value=" " />
  </exception>
  .....
</protected>
```

Figure 5.14: The Structure of Exceptions for Privacy Preferences

As a concrete example, you want your boss to access your location information only at week days from 9.00 to 18.00 and your wife to access it anytime unless you are within Germany. The relevant preferences can be expressed as in Figure 5.15. Similarly, if you want to reveal your interests only to persons whose interests are similar as yours, the relevant privacy preferences can be defined as in Figure 5.16.

```
<protected property="location">
  <exception role="family" id="wife">
    <if context="location" condition="is-not" value="Germany" />
  </exception>
  <exception role="work" id="boss">
    <if context="time" condition="is-not" value="weekend" />
    <if context="time" condition="between" value="09.00-18.00" />
    <if context="location" condition="is-not" value="Germany" />
  </exception>
</protected>
```

Figure 5.15: A Sample of Privacy Preferences for Location

```
<protected property="interest">
  <exception role="*">
    <if context="partner_interest" condition="similar"
      value=$own_interest />
  </exception>
</protected>
```

Figure 5.16: A Sample of Privacy Preferences for Interests

### Trust Management with P3P

After releasing your personal data to a service provider, you cannot control whether your data is misused or not. It can then be used for profiling, forwarded to other parties, used for spamming, etc. You need some trust relation with your partners before you release your data. With P3P [51], you can build this trust relation with your partners. P3P does not guarantee the enforcement of the policies, but it can be evaluated as a promise of providers.

Mobile identity management systems should be equipped with P3P support and its preference language should be extended as explained in the previous section. Integration of P3P avoids the pitfalls of “obscuring actual and potential information flow” and “lacking coarse-grained control”. Moreover, before any data is released to service providers, mobile users are asked to give permission or not for the transmission. This is an extra mechanism for avoiding the relevant pitfalls.

### Status as Soft Shut-Down Button

The privacy requirements of users are directly related to the status and mood of users. If a user is busy or away, he would not want to interact with any application. Similarly, if he is very angry or upset, he might refuse to take part in any mobile service. Therefore, a *status* option is a necessity for the mobile identity management system of context-aware applications. If a mobile user switches his status from online to offline, any data release should be automatically stopped.

The avoidance of the pitfall “lacking coarse-grained control” requires a simple mechanism to cut off the transfer of data, and with a simple status option, this can be achieved in the mobile identity management system for the Friend Finder application.

### History Management

History management in mobile identity management systems allows mobile users to follow their past activities (i.e. the released context data, date, time, the corresponding partner, etc.). In addition, history management allows users to interact directly with the receivers of their personal data, inform them about the cancellation of their consent and make them delete their personal data from their media.

Integrating history management within the Friend Finder’s identity management system, mobile users would be able check at any time to whom they have given away their location information. Therefore, the pitfalls “obscuring potential and actual information flows” and “inhibiting established practice” are avoided.

### Confidential Data Management

With the increasing popularity of mobile devices, confidentiality of mobile data has become more critical. Many mobile devices (e.g. laptops, PDAs, mobile phones) are either forgotten in taxis or public transport or they are stolen. If the data stored on mobile devices are not encrypted, the confidentiality of personal data is compromised.

Mobile identity management systems deal with personal data that are normally very sensitive. Therefore, mobile identity management system should apply encryption techniques “on the fly” and prevent illegal access of unauthorized people to confidential information. This aspect is related to privacy patterns and thus avoids the pitfall “inhibiting established practice”.

### Content and Communication Anonymity

Content anonymity requires staying anonymous at the application level. Pseudonyms can be used for enforcing content anonymity. Communication anonymity is related to network level anonymity. If a user communicates directly with a service provider, he leaves many signs that can be used for revealing the real identity of the user<sup>6</sup>. Communication anonymity networks and the relevant tools [62, 32, 106] today exist for preventing service providers from identifying users.

Mobile identity management systems should be equipped with such tools and pseudonym-support and enable users to communicate anonymously both at application and network level. If this is provided, the pitfall “inhibiting established practice” is avoided. Additionally, the mobile identity management hides the complexity of enforcing anonymity from users and therefore avoids the pitfall of “emphasizing configuration over action”.

#### 5.3.5 Integration of the Aspects

In this section, the aspects explained in the previous section are integrated within the Friend Finder application. The integration is illustrated in Figure 5.17. In the example, the location provider as a trusted party exists and collects location information from mobile users. Mr. Fischer, as a mobile user, allows his wife and his boss to query his location and track his movements on a visual map displayed on their mobile devices. The map provider receives the location information of mobile users from the location provider and presents them on a visual map for the mobile users.

The aspects in this application scenario are as follows: Before communicating with the location provider, mobile users specify which static and dynamic data are sent to the location provider. As shown in the Figure 5.17, Mr. Fischer releases his location information at the *city level*. This is the integration of the aspect “*blurring in levels*”. Additionally, he can push the button “Manage Appel Prefs.” and specify his Appel privacy preferences for the trust management with the location provider and map provider. He can also specify some exceptions related to the aspect *context-to-context relations* such as “my location information should be sent to my boss only on weekdays from 09.00 to 18.00” or “my location information should be sent to my wife on any day when I am in Germany”. In order to express such relations, an extensible preference language is needed. These are the inte-

---

<sup>6</sup>For a complete list of data revealed in case of direct communication, refer to <http://gemal.dk/browserspy/>



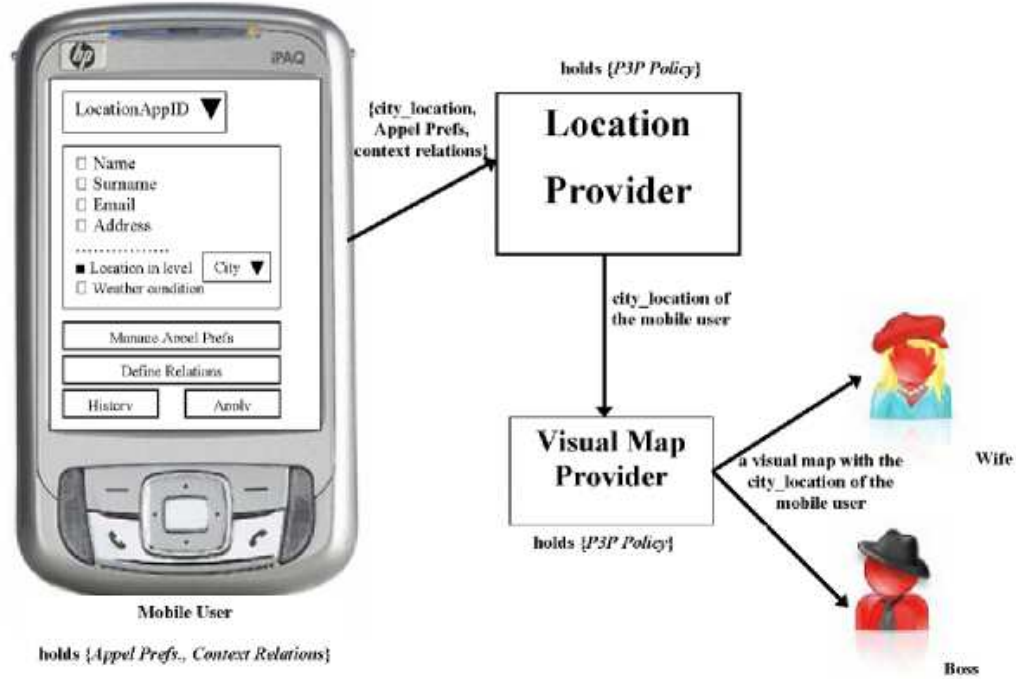


Figure 5.17: The Aspects integrated in the Friend Finder Application

gration of the aspects *context-to-context relation* and *extensible preference language*. Mr. Fischer can also access his history data summarizing what kind of information was released, at what time and to whom by pressing the “History” button. This is the “*history management*” aspect.

Afterwards, Mr. Fischer presses the “Apply” button and the communication with the location provider starts. Initially, the P3P policy of the location provider is retrieved and compared with his Appel preferences (i.e. the aspect *trust management with P3P*). If there is no conflict, he is asked to confirm that his location data will be sent periodically. At this point, the aspect “context-to-context dependence” comes into play. Mr. Fischer did not choose the attribute “weather condition” to be released. However, since the current location as the city name is released, it is also clear that the receiver can easily find out this attribute. He is warned against this conflict. If he confirms, his location is sent, within a set period of time,

to the location provider. Initially, he also sends his Appel preferences and context relation rules. The location provider takes his preferences and the exceptions into consideration and evaluates them before his location data is forwarded to other principals.

When the visual map provider asks for his location information, the location provider compares his Appel preferences and the P3P policy of the visual map provider. It also checks the exceptions for the context relations and then releases the relevant data to the visual map provider. In addition, if Mr. Fischer communicates directly with the map provider, the communication is built upon an anonymous network automatically (i.e. the aspect *anonymity*).

## Chapter 6

# Software-centric Proposed Solutions

### 6.1 Software Engineering for Security

It is a common saying within the security community that people are the weakest link in the security chain. This is also true for software engineering relating to security. The implementation of a security protocol can be secure if only the specification of the protocol is also secure. For example, after 17 years of its publication it was realized that Needham-Schroeder protocol [146] was not a secure protocol [135]. Since the specification is insecure, its implementations are also insecure.

However, secure specifications do not always result in secure implementations and this is mostly due to software developer errors. Recently, a very critical error was discovered in Debian openssl packages [13], for example. The problem was that the random number generator of Debian openssl was not sufficiently random, because one of the Debian developers wrongly changed a single line of code in September 2006 within the original openssl packages while trying to silence a warning message. As a result, SSH keys, OpenVPN keys, DNSSEC keys, key material for use in X.509 certificates and session keys used in SSL/TLS connections all affected by this bug and all Debian and Debian-based distributions had to be checked against weak key problems and the keys needed to be regenerated. This also shows that a small security mistake in software engineering can result in very critical failures. Moreover, people are the weakest link in the chain and it is sure more similar bugs will exist in our lives in the future.

Prevention of security bugs in applications has been studied for a long

time in academia and industry as well. One method is code review and penetration testing. Once the code is written, security experts can look for possible bugs by applying black and white box techniques [138] and try to apply countermeasure for the found bugs.

It is the case that testing applications and fixing bugs show only existence of bugs and not their non-existence. Other techniques are required to remove bugs before they appear in application code. Separation of codes [182, 126, 198] is a good technique to improve quality of application code in terms of security. In aspect-oriented programming [127], you separate functional business code from non-functional code (called *aspects*) (e.g. security, error handling, logging, etc.). In this case, security experts can focus only on the security aspects and application developers only need to take care of the business logic. Application developers can not easily modify security code, since they are physically separated. This improves the security development process.

However it should not be forgotten that security experts are also human beings, who can be in different psychological and physical situations and who can also code security bugs unintentionally. It is known that security bugs are very costly compared to simple application bugs. One further step in the prevention bugs due to security experts is the automation of code generation. If final source code can be generated from security specifications in an automated way, then it can be concluded that the security implementation is also secure<sup>1</sup>.

In the M-Business project, the cryptographic compiler LaCoDa (The Language for Code Generation and Protocol Analysis) which generates final source code from high level security protocol specifications has been implemented. It should be noted that the LaCoDa provides its own language for protocol specifications.

## 6.2 LaCoDa: The Cryptographic Compiler

The M-Business framework requires many security components and protocols integrated within service provider and mobile user applications. Application developers are not security experts – and even security experts might overlook certain security checks. As a result, implementing security functions from specifications is an error-prone task. LaCoDa (The Language for Code Generation and Protocol Analysis) is an attempt to automate the generation of security implementations from security specifications in order

---

<sup>1</sup>with the assumption that the code generator is also error-free.

to reduce or prevent the developer bugs which have an adverse effect on security.

The LaCoDa project provides both a specification language for security protocols and a compiler for generating final source code (however, currently only in Java) from specifications. The objectives were the creation of suitable specification language and then the design and development of a related compiler. As a result of analyzing the existing specification languages of verification tools (e.g. Capsl [141], Casper [114], Hlpsl [64], Laeva [193])<sup>2</sup>, it was concluded that they are not suitable for source code generation and the specification language and the compiler need to be designed from scratch.

The LaCoDa compiler project was initiated within the Master’s thesis namely “Design and Implementation of a cryptographic Compiler” [170] written by Nico Schmoigl. The compiler and the specification language were developed within this master thesis. Afterwards, the analysis and testing of the compiler has been completed within the Bachelor thesis namely “Test and further development of LaCoDa compiler” [125] written by Sylvie Kegne. Finally, adding new features and functionalities to the initial LaCoDa compiler has been carried out in a semester project work [83] completed by Andre Zoelitz, Xing Li and Juliane Lenz.

In this section, the architecture and the features of the specification language regarding LaCoDa are explained.

### 6.2.1 The Architecture

As illustrated in Figure 6.1, the LaCoda compiler consists of two parts – namely the front-end the back-end. The front-end receives high-level specifications of security protocols as input and generates tokens of the protocols into XML-based parse trees saved as external files. The back-end gets the parse trees and optionally the settings from template files, and finally generates source code for the relevant target platforms. The back-end is split into two parts – the Connector and Translator. The Connector enriches parse trees based on the information from templates (e.g. fixed key lengths, specific hash algorithms, etc.) or cryptographic object information. The Translator generates target platform specific code from enriched parse trees. To support more target platforms, one needs only to implement the relevant translators for LaCoDa. LaCoDa is designed to generate source code for different languages (e.g. Java, C, Ada), but currently it generates only Java code.

---

<sup>2</sup>For more information about the existing verification languages, refer to [170]/Section 2: Vorherige Arbeiten (in German)

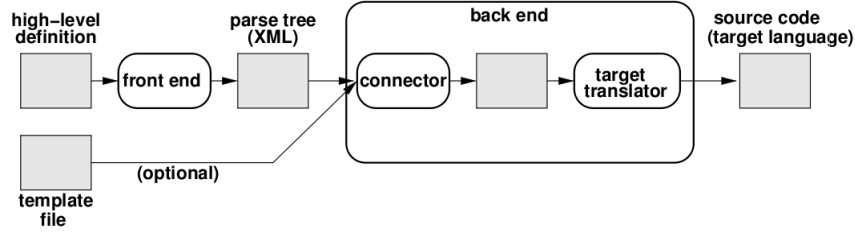


Figure 6.1: The Architecture of LaCoDa

### 6.2.2 The Specification Language

The LaCoDa specification language is object-oriented and utilizes cryptographic objects. There are two types of parameter definition, i.e. parameters and variables. Parameters (keyword *param*) are used for the communication with the environment, whereas variables (keyword *var*) store only internal computation results for temporary duration. Parameters can also be defined as shared (keyword *shared*). Public keys can be given as an example of shared parameters. A parameter can also be restricted with the keywords *in* and *out*, i.e. showing that a parameter is an inbound or outbound parameter respectively.

During protocol execution, it should be possible to check certain conditions and even fail if required. For example, if comparison of two hash values fails, the execution should be stopped. LaCoDa uses the keywords *assert* for compile time and *ensure* for run-time checks. The keyword *fail* triggers a halt in protocol execution.

*Template* types can be used to define abstract types during the design of protocols. Concrete definitions of primitives are retrieved from template files during compile time. As an example, *SymmetricChipherKey* in template type can be replaced with *AES256* from template file.

The *message* block in protocol specifications lists the exchanged messages between communication parties in order. While the specification language focuses on message-based view of protocols, final source code have rather role-based views.

Additionally, there are other certain features supported by the LaCoDa specification language:

- IF-THEN-ELSE: The concept of “IF-THEN-ELSE” is supported by the LaCoDa to check certain conditions and values during protocol

execution.

- Loops: “repeat-loops” is supported by the LaCoDa. The syntax of loops is as follows:

```
A,B: repeat
    A : x := rnd (...);
    A -> B : x -> y;
    until A:(x >3)
    until B:(y >3)
end repeat ;
```

- Import of libraries / Call of sub-protocols: For complex protocols, it is very useful to benefit from existing libraries and LaCoDa protocols. The features of “importing libraries” and “calling of sub-protocols” are supported by the LaCoDa.
- Functions: The features of “defining a function” and “calling a function” are supported by the LaCoDa.

For further information regarding the language notation, you can refer to the Appendix C which gives the language syntax in EBNF (Extended Backus-Naur-Form) form.

### 6.2.3 Template File

The high-level protocol specifications do not consider concrete cryptographic primitives and only defines them in an abstract way. “HashFunction” is more within its concern, rather than concrete MD5 or SHA1 functions. It is the case that the translator requires concrete values to accomplish its task. This gap is filled with the use of template files.

Another functionality of template files is that it enables dynamic replacement. If you need to replace SHA1 with SHA256, you do not need to make any changes within the specification, but only modify the relevant template file. In addition, you can specify certain lengths of primitives such as minimum key length, minimum hash length, etc.

In Figure 6.2, an example of template file is given. In this template file, the symmetric cipher is defined as AES with key size 128. The MAC primitive is defined as OMAC with key size 128 and MAC size 64.

```
S = AES
S.keysize = 128

M = OMAC
M.keysize = 128
M.Tag.size = 64
```

Figure 6.2: Template File Example

#### 6.2.4 Concrete Example: Encrypt-then-Authenticate Protocol

Encrypt-and-Authenticate is a protocol which guarantees confidentiality and authenticity of exchanged messages between two parties. In the protocol, both parties share one secret key for encryption and another secret key for the computation of message authentication code. The sender party encrypts the message and also computes the message authentication code (MAC) of the *enciphered message* and send both to the receiver. The receiver party firstly computes the MAC of the received cipher message and compares it with the MAC received. If they are identical, the authentication succeeds and the receiver also decrypts the message for further processing. The specification of this protocol with message flows in LaCoDa is shown in Figure 6.3.



```

system EncryptThenAuthenticate {
  template {
    S : SymmetricCipherKey;
    M : MacKey;
    assert(S, key_size >= 80);
    assert(M, key_size >= 80);
    assert(M.Tag, size >= 32);
  }
  param shared {
    in kEnc : S;
    in KAut : M;
  }
  param A {
    in B : Party;
    shared kEnc;
    shared KAut;
    in send_message : BitStream;
  }
  param B {
    in A : Party;
    shared kEnc;
    shared KAut;
    out receive_message : BitStream;
  }
  var A {
    ciphertext : BitStream;
  }
  var B {
    c_txt : BitStream;
    aut_tag : M.Tag;
  }
  messages {
    A      : ciphertext := kEnc.encrypt( send_message );
    A -> B : ciphertext -> c_txt;
    A -> B : KAut.MAC_generate(ciphertext) -> aut_tag;
    B      : if KAut.MAC_verify(c_txt, aut_tag) = 0 then
                fail(1);
  endif
    B      : receive_message := kEnc.decrypt(c_txt);
  }
}

```

Figure 6.3: The protocol specification of the Encrypt-then-Authenticate<sup>3</sup><sup>3</sup>Referenced from [170]

The specification can be interpreted as follows:

- The name of the protocol specification is `EncryptThenAuthenticate`.
- In the template block, minimum key lengths of symmetric cipher (i.e. `S`) and MAC (i.e. `M`), and also minimum length of MAC result are specified.
- In the param shared block, encryption (i.e. `kEnc`) and authentication (i.e. `kAut`) keys are defined as shared parameters.
- In the param A block, communication party (i.e. `B`) and the message to be sent (i.e. `send_message`) are defined as in parameters.
- In the param B block, communication party (i.e. `A`) as in parameter and the message to be received (i.e. `send_message`) as out parameter are defined.
- In the var A block, the computed encryption result (i.e. `ciphertext`) is defined.
- In the var B block, the received encryption result (i.e. `c_txt`) and the received MAC result (i.e. `aut_tag`) are defined.
- In the messages block, the computed and exchanged messages between communication parties are listed.

Based on the specification above and the template file in Figure 6.2, the LaCoDa compiler generates Java source code for Party A (i.e. `EncryptThenAuthenticate_A.java`) and Party B (i.e. `EncryptThenAuthenticate_B.java`) as shown in Appendix B.1 and B.2 respectively.

### 6.2.5 Implementing Security Protocols

For testing the flexibility of the specification language and the functionality of the compiler, Sylvie Kegne implemented some test cases using LaCoDa [125]. The following steps were applied for the test:

- Choosing the security protocols: 13 common security protocols were chosen from the SPORE (Security Protocols Open Repository) [178]. These protocols are: Ban modified Andrew Secure RPC, Loew modified Ban concrete Andrew Secure RPC, CCITT X509, Loew modified

Denning-Sacco shared key, Diffie-Hellmann, Kao Chow Authentication v3, Kerberos v5, Kehne-Schoenwaelder-Landendoerfer, Needham-Schroeder Symmetric Key, Shamir-Rivest-Adleman Three Pass, SmartRight view-only, Wired Equivalent Privacy and Woo-Lam Pi 3

- Encoding the protocols in LaCoDa: Each security protocol was encoded with the specification language of LaCoDa for examining the compiler.
- Running the compiler: For generating final Java code, the compiler was executed with the encoded security protocols and the compiler was analyzed.
- Compiling and running the final source code: In the final step, the source code generated by LaCoDa was compiled using a Java compiler and the generated class files were executed to test the data flow of the security protocols.

Based on the results from the tests, the following extensions were added to LaCoDa:

- Network Communication Layer: During the execution, the involved parties in the protocol should be able to exchange messages on the network layer. A socket-based communication layer was implemented and integrated into LaCoDa.
- New Objects at Compile-time: The compiler needs certain cryptographic objects (i.e. CO) files to generate final code from the specification. For example, if a hash function is used as a type in the specification, the relevant CO file should be available for the compiler. Hash, Timestamp, Checksum and Bignum object files were created and added to the compiler. The object Bignum is required by the Diffie-Hellman protocol. The object Nonce already existed within the LaCoDa, but add and subtract methods, which are required by some protocols like BAN modified Andrew Secure RPC, were not implemented.
- New Objects at Run-time: The specification language defines protocols in an abstract way. It requires the message to be exchanged between two parties to be encrypted, but does not specify which algorithm is to be used for encryption. For the concrete mapping of algorithms, templates are used. The specified algorithms should be

available to the compiler as well. As a result, AES for symmetric encryption and OMAC for message authentication code were implemented for the compiler.

### 6.2.6 Discussion

Even though LaCoDa is at an early stage, it has been successful in showing that is a powerful method for reducing the number of security bugs. The next step in the development of LaCoDa should be its integration within a larger number of software development processes. If this is done, there will be a need for further development – including enhancements to the compiler and the language itself.

## Chapter 7

# Conclusion and Future Work

The provision of automatic location determination, context-aware and location-aware applications has enabled a new trend in mobile business and mobile commerce transactions. The existing applications (e.g. mobile navigation, child tracking, automatic panic alarms and restaurant finders) have already brought additional benefits to the daily lives of many people by helping to overcome these complicated problems.

In contrast, security and privacy have become the biggest barriers against the long-term success of these applications, since mobile users and service providers are afraid of being disadvantaged by internal or external attackers. In this thesis, different aspects of security and privacy are studied as regards context-aware applications and from the perspectives of mobile users and service providers.

Chapter 1 gave an introduction into this topic, the M-Business research group at the University of Mannheim and the thesis. In Chapter 2, common terms and concepts that are used through the thesis are explained and a categorization of context-aware applications is given. Existing context-aware applications, the target application scenarios within the thesis and location determination techniques are also examined and explained in this chapter.

Chapter 3 introduced the principles of information security and gave the results of our security analysis of context-aware applications by detailing possible threats and solutions.

Chapter 4 explained the possible privacy challenges in a specific context-aware application (i.e. Friend Finder) and also the legal aspects based on EU data protection and e-privacy directives. To illustrate real life threats, privacy risks appeared in the media and Google hacking exploits (with possible countermeasures suggested) were also analyzed in this chapter.

In Chapter 5, user-centric solutions are explored and proposed. A client security architecture consisting of different security components was introduced. The architecture of the dynamic anonymity solution enabling policy-based anonymity for mobile users was explained in detail. The new aspects of context-aware applications in terms of mobile identity management and their integration within the M-Business framework concluded this chapter.

In Chapter 6, software-centric solutions are explored and proposed. This chapter was dedicated to our cryptographic compiler LaCoDa which generates final Java code from high level specifications of security protocols and thus minimizes or even prevents security bugs stemming from application developers.

Security and privacy measures are difficult to enforce, since they require not only technical countermeasures but also social countermeasures. On mobile platforms, the problem is more difficult to resolve and the consequences of failure more serious. However we expect that mobile devices will be improved and will provide more performance. They will be capable of implementing highly sophisticated security techniques. Security requires education of people. Real life security failures (e.g. online banking frauds) often appear in the media and people are becoming more aware of possible threats and are becoming more sensitive as to their personal data security and privacy. New security threats are, however, constantly being discovered and new risks are becoming apparent. Therefore more researchers are expected to explore the mobile security domain with a focus on service-oriented architectures.

# Bibliography

- [1] AdvancedDork-A Firefox plug-in for advanced Google Search.  
<https://addons.mozilla.org/en-US/firefox/addon/2144>.
- [2] AOL phisher gets seven year sentence.  
[http://www.pcworld.com/businesscenter/article/149790/aol\\_phisher\\_gets\\_seven\\_year\\_sentence.html](http://www.pcworld.com/businesscenter/article/149790/aol_phisher_gets_seven_year_sentence.html).
- [3] Appel: A P3P Preference Exchange Language.  
<http://www.w3.org/TR/P3P-preferences/>.
- [4] ATM locators. <http://www.postank.de/mobileservices>.
- [5] Auto search service. <http://www.ermittlungenallerart.de>.
- [6] AxCrypt File Encryption Software for Windows.  
<http://axcrypt.axantum.com>.
- [7] Blind Guidance System. <http://www.navtec.de>.
- [8] Blog of Ellen Simonetti. <http://queenofsky.journalspace.com/>.
- [9] BrowserSpy. <http://gemal.dk/browserspy/>.
- [10] cryptlib Encryption Toolkit.  
<http://www.cs.auckland.ac.nz/pgut001/cryptlib/>.
- [11] Cyber crime news. <http://cybercrimeupdates.blogspot.com/>.
- [12] Datenschutzskandal bei Krankenkasse (in German).  
[http://www.computerbild.de/artikel/cb-News-Sicherheit-Report-Datenschutzskandal-bei-Krankenkasse\\_3249331.html](http://www.computerbild.de/artikel/cb-News-Sicherheit-Report-Datenschutzskandal-bei-Krankenkasse_3249331.html).
- [13] Debian Security Advisory, DSA-1571-1 openssl – predictable random number generator. <http://www.debian.org/security/2008/dsa-1571>.

- [14] DFG-Deutsche Forschungsgemeinschaft. <http://www.dfg.de/en/>.
- [15] Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).  
[http://www.dataprotection.ie/documents/legal/directive2002\\_58.pdf](http://www.dataprotection.ie/documents/legal/directive2002_58.pdf).
- [16] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.  
[http://www.cdt.org/privacy/eudirective/EU\\_Directive\\_.html](http://www.cdt.org/privacy/eudirective/EU_Directive_.html).
- [17] Facebook-a social network connecting people.  
<http://www.facebook.com>.
- [18] FIDIS (Future of Identity in the Information Society).  
<http://www.fidis.net>.
- [19] FIDIS Project - Database on Identity Management Systems.  
<http://www.fidis.net/interactive/ims-db/>.
- [20] Fleet management. <http://www.fleetonline.ch>.
- [21] Galileo - European Satellite Navigation System.  
[http://ec.europa.eu/dgs/energy\\_transport/galileo/index\\_en.htm](http://ec.europa.eu/dgs/energy_transport/galileo/index_en.htm).
- [22] GEM Project-Generic Environment for Mobile Business.  
<http://www.m-business.uni-mannheim.de/GEM/Home.htm>.
- [23] Google Advanced Search. [http://www.google.com/advanced\\_search?hl=en](http://www.google.com/advanced_search?hl=en).
- [24] Google Hacking. [http://en.wikipedia.org/wiki/Google\\_hacking](http://en.wikipedia.org/wiki/Google_hacking).
- [25] Google Hacking Database. <http://johnny.ihackstuff.com/ghdb.php>.
- [26] Goolink- Security Scanner.  
[www.ghacks.net/2005/11/23/goolink-scanner-beta-preview/](http://www.ghacks.net/2005/11/23/goolink-scanner-beta-preview/).
- [27] Group management via tracking of group members.  
<http://www.sintrade.ch>.
- [28] Herecast- an open infrastructure for location-based services using WiFi. <http://www.herecast.com/services/friendfinder/>.



- [29] idemix-a tool for pseudonymity for e-transactions.  
<http://www.zurich.ibm.com/security/idemix>.
- [30] Indoor & outdoor routing. <http://www.falk.de>.
- [31] Indoor navigation in fairs. <http://www.cebit.de>.
- [32] Jap: Anonymity and Privacy Tool for Internet.  
<http://anon.inf.tu-dresden.de>.
- [33] Kerberos: The Network Authentication Protocol.  
<http://web.mit.edu/Kerberos/>.
- [34] Kid tracking. <http://www.trackyourkid.de>.
- [35] LAMBADA Project - Location-Aware Mobile Business Adhoc Architecture. <http://www.m-business.uni-mannheim.de/LAMBADA/Home.htm>.
- [36] Landesstiftung Baden-Württemberg.  
<http://www.landesstiftung-bw.de>.
- [37] Localization of drivers having car break-downs and accidents.  
[www.notfon-d.de](http://www.notfon-d.de).
- [38] Localization of friends, relatives and family members.  
<http://www.mobiloco.de>.
- [39] Locating people in emergency. <http://www.sintrade.ch>.
- [40] Location-based chat and games. <http://www.vodafone.de>.
- [41] Location-based game. <http://www.mogimogi.com>.
- [42] Location-based Services Application Prototype.  
<https://www.prime-project.eu/prototypes/lbs/>.
- [43] Mobile city guide. <http://www.al.com/mobile/cityguide/>.
- [44] Mobile phone search service. <http://www.o2online.de>.
- [45] Navigation to restaurants and shopping centers.  
<http://www.vindigo.com>.
- [46] Navigation to restaurants and shopping centers.  
<http://www.mobiloco.de>.

- [47] The Nexus project.  
<http://www.nexus.uni-stuttgart.de/index.en.html>.
- [48] The Nimbus project.  
<http://www.wireless-earth.de/nimbus.html>.
- [49] OASIS Web Services Security. [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss).
- [50] OpenLS - Location Services. <http://www.openls.org>.
- [51] P3P (Platform for Privacy Preferences). <http://www.w3.org/P3P/>.
- [52] Phishing attacks on Tor anonymisation network. <http://www.heise-security.co.uk/news/95778>.
- [53] PRIME - Privacy and Identity Management for Europe.  
<http://www.prime-project.eu>.
- [54] Robots Exclusion Standard. <http://en.wikipedia.org/wiki/Robots.txt>.
- [55] RSA secureid cards for two-factor authentication.  
<http://www.rsa.com/node.aspx?id=1156>.
- [56] SALSA Project-Software Architectures For Location-Specific Transactions in Mobile Commerce.  
<http://www.m-business.uni-mannheim.de/SALSA/Home.htm>.
- [57] Seattle Dark Mailer faces 47-month sentence.  
[http://www.theregister.co.uk/2008/07/23/soloway\\_sentenced/](http://www.theregister.co.uk/2008/07/23/soloway_sentenced/).
- [58] Simple Public Key Infrastructure.  
<http://world.std.com/~cme/html/spki.html>.
- [59] Status of Jap cascades. <http://anon.inf.tu-dresden.de/status.php>.
- [60] Taking steps to further improve our privacy practices.  
<http://googleblog.blogspot.com/2007/03/taking-steps-to-further-improve-our.html>.
- [61] Teen hacker could get 38-year sentence for fixing grades.  
<http://www.technewsworld.com/story/security/63483.html?wlc=1221316903>.
- [62] The Anonymizer. <http://www.anonymizer.com>.

- 
- [63] The Gnu Privacy Guard. <http://www.gnupg.org>.
  - [64] The High Level Protocol Specication Language: AVISPA-Project IST-2001-39252, Deliverable D2.1. <http://www.avispa-project.org/delivs/2.1/d2-1.pdf>.
  - [65] The M-Business Research Group Workshop. <http://www.m-business.uni-mannheim.de/workshopMBusiness/mBusinessWorkshop05.htm>.
  - [66] The Mobile Business Research Group - University of Mannheim. <http://www.m-business.uni-mannheim.de>.
  - [67] The Nidaros Framework for Location-aware Applications. [www.idi.ntnu.no/grupper/su/publ/alfw/mobis2005-nidaros-framework.pdf](http://www.idi.ntnu.no/grupper/su/publ/alfw/mobis2005-nidaros-framework.pdf).
  - [68] The Open Geospatial Consortium. <http://www.opengeospatial.org>.
  - [69] The User Agent Profile (UAProf). <http://en.wikipedia.org/wiki/UAProf>.
  - [70] Tracking of seniors/persons in need. <http://www.sintrade.ch>.
  - [71] Traffic Information. <http://www.verkehrsinfo.de>.
  - [72] Tripwire. <http://www.tripwire.org>.
  - [73] UAProf (User Agent Profile) Specification. <http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf>.
  - [74] The WASP project. <http://www.freeband.nl/kennisimpuls/projecten/wasp/ENindex.html>.
  - [75] Weather Information. <http://www.wetter.de>.
  - [76] XING-relationships for the worlds business professionals. <http://www.xing.com>.
  - [77] ISO99 IS 15408. <http://www.commoncriteriaportal.org>, 1999.
  - [78] Tough penalties for mobile phone theft. *BBC News*, 3. May 2002.

- [79] SiteDigger v2.0 - Information Gathering Tool.  
<http://www.foundstone.com/us/resources/proddesc/sitedigger.htm>,  
June 2005.
- [80] Five ways to delete your google cookie.  
[http://googlewatch.eweek.com/content/five\\_ways\\_to\\_delete\\_your\\_google\\_cookie.html](http://googlewatch.eweek.com/content/five_ways_to_delete_your_google_cookie.html), July 2006.
- [81] Geographic location/privacy (geopriv).  
<http://www.ietf.org/html.charters/geopriv-charter.html>, September 2006.
- [82] Google Hack HoneyPot Project. <http://ghh.sourceforge.net>, 2007.
- [83] Implementation of security protocols. <http://www.uni-weimar.de/cms/medien/mediensicherheit/teaching/former-semester/implementation-von-sicherheitsprotokollen.html>, 2007.  
Bauhaus-University of Weimar.
- [84] OASIS WS-SecureConversation 1.3. <http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html>,  
1. March 2007.
- [85] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-Awareness. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307, London, UK, 1999. Springer-Verlag.
- [86] M. Ackerman, T. Darrell, and D. Weitzner. Privacy In Context. *The journal of Human-Computer Interaction*, 16(2-4), 2001. Special Issue on Context-Aware Computing.
- [87] R. Anderson. *Security Engineering*. Wiley Computer Publishing, 2001.
- [88] Christer Andersson, Reine Lundin, and Simone Fischer-Hübner. Privacy-enhanced WAP Browsing with mCrowds - Anonymity Properties and Performance Evaluation of the mCrowds System. In *Proceedings of the Fourth annual ISSA 2004 IT Security Conference*, pages 85–90, Johannesburg, July 2004.
- [89] P. Ashley, S. Hada, G. Karjoth, and M. Schunter. E-P3P privacy policies and privacy authorization. In *Proceedings of the ACM workshop*

- on Privacy in the Electronic Society (WPES 2002)*, pages 103–109. ACM Press, 2002.
- [90] P. Ashley, S. Hada, C. Powers, and M. Schunter. Enterprise Privacy Authorization Language (EPAL). Technical Report 3485, IBM Research, 2003.
- [91] B. Koelmel. Location-based Services-Eine Killerapplikation für UMTS. *www.e-lba.com*, 2004.
- [92] B. Rao and L. Minakais. Evolution of mobile location-based services. In: *Communications of the ACM*, 46. Jg. (2003), Nr. 12, pages 61–65, 2003.
- [93] Hagen Barlag and Stephan Drautz. Concept and implementation of security architecture (in german). Master’s thesis, University of Hagen, October 2003.
- [94] S.J. Barnes. Location-Based Services The State of the Art. In *E-Service Journal*, 2. Jg. (2003), Nr. 3, pages 59–70, 2003.
- [95] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against Tor. In *WPES ’07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 11–20, New York, NY, USA, 2007. ACM.
- [96] Christian Beil. Development of a Framework for Dynamic Mobile Anonymity. Bachelor Thesis, University of Mannheim, December 2005.
- [97] Alastair R. Beresford. Location privacy in Ubiquitous Computing. Technical Report UCAM-CL-TR-612, University of Cambridge, Computer Laboratory, January 2005.
- [98] Raheem Beyah, Shantanu Kangude, George Yu, Brian Strickland, and John Copeland. Rogue Access Point Detection using Temporal Traffic Characteristics. In *Proceedings of IEEE GLOBECOM*, December 2004.
- [99] Matt Bishop. *Introduction to Computer Security*. Pearson Education, Boston, MA, 2005.
- [100] Bouncy Castle Crypto APIs. <http://www.bouncycastle.org>.

- [101] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven A. Shafer. EasyLiving: Technologies for Intelligent Environments. In *2nd International Symposium on Handheld and Ubiquitous Computing (HUC'00)*, pages 12–29, Bristol, UK, 2000. Springer-Verlag.
- [102] D. Chaum. The dining cryptographers problem: unconditional sender and recipient untraceability. *J. Cryptol.*, 1(1):65–75, 1988.
- [103] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.
- [104] A. Dey. Understanding and using context. In *Personal and Ubiquitous Computing, Vol 5, No. 1*, pages 4–7, 2001.
- [105] N. Diezmann. *Report Mobile Business - Neue Wege zum mobilen Kunden*, chapter Payment - Sicherheit und Zahlung per Handy (in German), pages 155–178. 2001.
- [106] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.
- [107] E. Isaacs and A. Walendowski and D. Ranganathan. Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. in: *CHI 2002 Conference Proceedings. Minneapolis. Minnesota. USA*.
- [108] Ginger Myles et al. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.
- [109] S. L. Jarvenpaa et al. Mobile Commerce at Crossroads. *Communications of the ACM*, 46(12):41–44, 2003.
- [110] Niels Ferguson and Bruce Schneier. *Practical Cryptography*, chapter 22: Storing Secrets, pages 357–358. John Wiley and Sons, Inc., 2003.
- [111] Simone Fischer-Hübner, M. Nilsson, and Helena Lindskog. Self-Determination in Mobile Internet: PiMI Prototype Results. In *SEC '02: Proceedings of the IFIP TC11 17th International Conference on Information Security*, pages 373–386, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V.

- [112] Michael J. Freedman and Robert Morris. Tarzan: A Peer-to-Peer Anonymizing Network Layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.
- [113] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.
- [114] Phillippa Broadfoot und Mei Lin Hui Gavin Lowe. CASPER: A Compiler for the Analysis of Security Protocols, User Manual and Tutorial. <http://web.comlab.ox.ac.uk/oucl/work/gavin.lowe/Security/Casper/manual.ps>, Dezember 2001. Version 1.5.
- [115] Giles Hogben - Suggestions for long term changes to P3P. W3C Workshop on the long term Future of P3P and Enterprise Privacy Languages, 2003.
- [116] Oded Goldreich. A Tutorial about Zero-Knowledge, March 2004. <http://www.wisdom.weizmann.ac.il/~oded/zk-tut02.html>.
- [117] Dieter Gollmann. *Computer Security*, chapter 13: Network Security, pages 232–235. John Wiley and Sons, Ltd., 1st edition, 1999.
- [118] Andreas Gutscher. Coordinate transformation - a solution for the privacy problem of location-based services. In *IPDPS*, 2006.
- [119] Hans H. Bauer and Tina Reichardt and Anja Schüle. Was will der mobile Nutzer? Forschungsergebnisse zu den Anforderungen von Nutzern an kontextsensitive Dienste (in German). *Haasis, K./Heinzl, A./Klumpp, D. (Hrsg., 2006): Aktuelle Trends in der Softwareforschung, Heidelberg*, pages 179–191, 2006.
- [120] Jason I. Hong and James A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189, New York, NY, USA, 2004. ACM.
- [121] A. Jagoe. Mobile Location Services. *Upper Saddle River*, page 119, 2003.
- [122] Uwe Jendricke and Daniela Gerd tom Markotten. Usability meets Security - The Identity-Manager as your Personal Security Assistant

- for the Internet. In *Proceedings of the 16th Annual Computer Security Applications Conference*, pages 344–353, December 2000.
- [123] Jörg Link and Sebastian Schmidt. Erfolgsplanung und kontrolle im Mobile Commerce (in German). In: *Silberer, Günter / Wohlfahrt, Jens/ Wilhelm, Torsten (Hrsg.), Mobile Commerce. Grundlagen, Geschäftsmodelle, Erfolgsfaktoren*, pages 128–149, 2002.
- [124] Elliott Kaplan. *Understanding GPS (Global Positioning System): Principles and Applications*. Artech House, Inc., 2nd edition, December 2005.
- [125] Sylvie Kegne. Test und Weiterentwicklung des LaCodA Compilers (in German). Bachelor Thesis, University of Mannheim, May 2006.
- [126] Gregor Kiczales, John Lamping, Anurag Menhdhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin. Aspect-Oriented Programming. In Mehmet Aksit and Satoshi Matsuoka, editors, *Proceedings European Conference on Object-Oriented Programming*, volume 1241, pages 220–242, Berlin, Heidelberg, and New York, 1997. Springer-Verlag.
- [127] Gregor Kiczales and Mira Mezini. Separation of Concerns with Procedures, Annotations, Advice and Pointcuts. In *European Conference on ObjectOriented Programming (ECOOP)*, 2005.
- [128] Thomas King, Thomas Haenselmann, Stephan Kopf, and Wolfgang Effelsberg. Technical Report: Positionierung mit Wireless-LAN und Bluetooth. Technical report, Department for Mathematics and Computer Science, University of Mannheim, 12. December 2005.
- [129] Stefan Köpsell, Hannes Federrath, and Marit Hansen. Erfahrungen mit dem Betrieb eines Anonymisierungsdienstes (in German). *Datenschutz und Datensicherheit*, 27(3), 2003.
- [130] Marc Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In *Proceedings of the 4th International Conference on Ubiquitous Computing*, pages 237–245, London, UK, 2002. Springer-Verlag.
- [131] Scott Lederer, I. Hong, K. Dey, and A. Landay. Personal privacy through understanding and action: five pitfalls for designers. *Personal Ubiquitous Computing*, 8(6):440–454, 2004.



- [132] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS Protocol Version 5. RFC1928, March 1996.
- [133] Johnny Long. Gooscan Google Security Scanner.  
[http://johnny.ihackstuff.com/downloads/task,doc\\_download/gid,28/](http://johnny.ihackstuff.com/downloads/task,doc_download/gid,28/).
- [134] Johnny Long. *Google Hacking for Penetration Testers*. Syngress Publishing Inc., Rockland, MA, 2005.
- [135] Gavin Lowe. An Attack on the Needham-Schroeder Public-Key Authentication Protocol. *Information Processing Letters*, 56(3):131–133, 1996.
- [136] David J. MacDonald. NTT DoCoMo's i-mode: Developing win-win relationships for mobile commerce. In *B. E. Mennecke & T. Strader (Eds.), Mobile Commerce: Technology, theory, and applications*, pages 1–25, 2003.
- [137] Martin Keßler. Tracking Dog - Implementation of a penetration testing tool for searching cryptographic secrets and personal secrets with Google. Bachelor Thesis, Bauhaus University of Weimar, Faculty of Media, October 2007.
- [138] Gary McGRAW. *Software Security*. Pearson Education, Inc., 1st edition, 2006. pg. 89-91.
- [139] Robert McMillan and IDG News Service. Google now a hacker's tool.  
[http://www.infoworld.com/article/05/08/02/HNgooglehackertool\\_1.html](http://www.infoworld.com/article/05/08/02/HNgooglehackertool_1.html), 02 August 2005.
- [140] Robert McMillan and IDG News Service. Corporate data slips out via Google calendar. [http://www.infoworld.com/article/07/04/17/HN-googlecalendardata\\_1.html](http://www.infoworld.com/article/07/04/17/HN-googlecalendardata_1.html), 17 April 2007.
- [141] J. Millen and F. Muller. Cryptographic Protocol Generation from CAPSL. Technical Report SRI-CSL-01-07, SRI International, December 2001.
- [142] Robert P. Minch. Privacy Issues in Location-Aware Mobile Devices. In *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 5*, page 50127.2, Washington, DC, USA, 2004. IEEE Computer Society.

- [143] Jelena Mirkovic, Sven Dietrich, David Dittrich, and Peter Reiher. *Internet Denial of Service: Attack and Defense Mechanisms*. Prentice Hall PTR, 2004. ISBN:0131475738.
- [144] Dirk Möhlenbruch and Ulf-Marten Schmieder. Mobile Marketing als Schlüsselgrösse für Multichannel-Commerce (in German). In: *Silberer, Günter / Wohlfahrt, Jens/ Wilhelm, Torsten (Hrsg.), Mobile Commerce. Grundlagen, Geschäftsmodelle, Erfolgsfaktoren*, pages 64–86, 2002.
- [145] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. Draft, July 2003.
- [146] Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *CACM*, 21:993–999, December 1978.
- [147] Alexander T. Nicolai and Thomas Petersmann. Der Möglichkeitsraum des Mobile Business eine qualitative Betrachtung (in German). In: *Strategien im M-Commerce, ed. by Nicolai, A.; Petersmann, T., Stuttgart (Schäffer-Poeschel)*, pages 19–21, 2001.
- [148] Mikael Nilsson, Helena Lindskog, and Simone Fischer-Hübner. Privacy Enhancements in the Mobile Internet. In *Proceedings of the IFIP WG 9.6/11.7 working conference on Security and Control of IT in Society*, Bratislava, June 2001.
- [149] Times Online. Shops track customers via mobile phone. [http://technology.timesonline.co.uk/tol/news/tech\\_and\\_web/article3945496.ece](http://technology.timesonline.co.uk/tol/news/tech_and_web/article3945496.ece), 16 May 2008.
- [150] Andreas Pfitzmann and Marit Hansen. Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology - Version v0.31, 15 February 2008.
- [151] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. ISDN-mixes: Untraceable communication with very small bandwidth overhead. In *Proceedings of the GI/ITG Conference on Communication in Distributed Systems*, pages 451–463, February 1991.
- [152] Oscar Pozzobon, Chris Wullems, and Prof. Kurt Kubik. Security issues in next generation satellite systems.

- [http://radio.feld.cvut.cz/satnav/CGSIC/presentations/DAY\\_1\\_am/Pozzobon\\_CGSIC\\_prague\\_final.ppt](http://radio.feld.cvut.cz/satnav/CGSIC/presentations/DAY_1_am/Pozzobon_CGSIC_prague_final.ppt), CGSIC Meeting, Prague, Czech Republic, 2005.
- [153] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis. On indoor position location with wireless LANs, Telecommunications Program & Dept. of Computer Science, University of Pittsburgh, PA., 2001.
- [154] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *MobiCom '00: 6th annual international conference on Mobile computing and networking*, pages 32–43, Boston, Massachusetts, United States, 2000. ACM Press.
- [155] Prof. Dr. Armin Heinzl. Chair of Business Administration and Information Systems, University of Mannheim. <http://wifo1.bwl.uni-mannheim.de/>.
- [156] Prof. Dr. Colin Atkinson. Chair of Software Technology, University of Mannheim. <http://swt.informatik.uni-mannheim.de>.
- [157] Prof. Dr. Dr. Martin Schader. Chair of Information System III, University of Mannheim. <http://schader.bwl.uni-mannheim.de/1/de/index.html>.
- [158] Prof. Dr. Guido Moerkotte. Chair of Database Technology, University of Mannheim. <http://pi3.informatik.uni-mannheim.de/>.
- [159] Prof. Dr. Hans H. Bauer. Chair of Business Administration and Marketing II, University of Mannheim. <http://bauer.bwl.uni-mannheim.de/>.
- [160] Prof. Dr. Matthias Krause. Chair of Theoretical Computer Science, University of Mannheim. <http://th.informatik.uni-mannheim.de/>.
- [161] Prof. Dr. Wolfgang Effelsberg. Chair of Multimedia and Network Technology, University of Mannheim. <http://www.informatik.uni-mannheim.de/pi4/>.
- [162] JR Raphael. Cell Phone Spying: Is Your Life Being Monitored? <http://www.geeksaresexy.net/2008/05/05/cell-phone-spying-is-your-life-being-monitored/>, 05 May 2008.
- [163] Michael Reiter and Aviel Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

- [164] Phillip Rogaway. Authenticated-encryption with associated-data. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 98–107. ACM Press, 2002.
- [165] Deborah Russell and G.T. Gangemi Sr. *Computer Security Basics*. O'Reilly & Associates, Inc., 1992.
- [166] S. L. Jarvenpaa et al. Mobile Commerce at Crossroads. in: *Communications of the ACM. Vol. 46 (2003). No. 12.*, pages 41–44, 2003.
- [167] S. Schwiderski-Grosche and H. Knospe. Secure Mobile Commerce. *Electronics Communications Engineering Journal: Special issue security for mobility*, 14(5):228–238, October 2002.
- [168] Bill Schilit, Norman Adams, and Roy Want. Context-Aware Computing Applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 85–90, Santa Cruz, CA, U, December 1994.
- [169] Albrecht Schmidt, Michael Beigl, and Hans-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.
- [170] Dominic Schmoigl. Design und Implementation eines kryptographischen Compilers (in German). Master's thesis, University of Mannheim, June 2005.
- [171] Bruce Schneier. A self-study course in block-cipher cryptanalysis. <http://www.schneier.com/paper-self-study.pdf>.
- [172] Bruce Schneier. Dan egerstad arrested. *Blog of Schneier on Security*, 16 November 2007. [http://www.schneier.com/blog/archives/2007/11/dan\\_egerstad\\_ar.html](http://www.schneier.com/blog/archives/2007/11/dan_egerstad_ar.html).
- [173] Bruce Schneier. *Secrets and Lies—Digital Security in a Networked World*, chapter 6: Cryptography, page 85. John Wiley and Sons, Inc., 2000.
- [174] Bruce Schneier. *Secrets and Lies—Digital Security in a Networked World*, chapter 5: Security Needs, pages 73–77. John Wiley and Sons, Inc., 2000.
- [175] James Sherwood. Teens use technology to party in strangers' pools. [http://www.reghardware.co.uk/2008/06/18/tech\\_aids\\_pool\\_crashing/](http://www.reghardware.co.uk/2008/06/18/tech_aids_pool_crashing/), 18 June 2008.

- [176] Ellen Simonetti. I was fired for blogging. *CNET News*, 16 December 2004. [http://www.news.com/I-was-fired-for-blogging/2010-1030\\_3-5490836.html](http://www.news.com/I-was-fired-for-blogging/2010-1030_3-5490836.html).
- [177] Sarah Spiekermann. *Location-based Services*, chapter General Aspects of Location-Based Services. Morgan Kaufmann, 2004.
- [178] SPORE. Security Protocols Open Repository. <http://www.lsv.ens-cachan.fr/spore/index.html>.
- [179] Polly Sprenger. Sun on privacy: Get over it. *Wired Magazine*, 26 January 1999. <http://www.wired.com/politics/law/news/1999/01/17538>.
- [180] Mark Stamp. *Information Security Principles and Practice*. John Wiley & Sons, Inc., 2006. pg. 2-3.
- [181] Stefan Lucks, Nico Schmoigl and Emin Islam Tatlı. Issues on Designing a Cryptographic Compiler. In *WEWoRC (Western European Workshop on Research in Cryptology)*, Leuven-Belgium, 2005.
- [182] Emin Islam Tatlı. Separation of Business and Security Logic. Master's thesis, Albert-Ludwigs University of Freiburg (in cooperation with SAP AG Corporate Research-Karlsruhe), 2004.
- [183] Emin Islam Tatlı. Context Data Model for Privacy. In *PRIME Standardization Workshop*, IBM Zürich Research Center, July 2006.
- [184] Emin Islam Tatlı. Google reveals Cryptographic Secrets. Technical Report of 1. Crypto Weekend, Kloster Bronbach, Germany, July 2006.
- [185] Emin Islam Tatlı. Extending P3P/Appel for Friend Finder. In *The International Workshop on Privacy-Aware Location-based Mobile Services (PALMS07)*, May 2007.
- [186] Emin Islam Tatlı. Google Hacking for Privacy. Third International Summer School The Future of Identity in the Information Society, Karlstad-Sweden, 6-10 August 2007.
- [187] Emin Islam Tatlı. Privacy in Danger: Let's google your privacy. In Albin Zuccato Leonardo Martucci Simone Fischer-Hübner, Penny Duquenoy, editor, *In Proceedings of the Third IFIP WG 9.2, 9.6/11.6, Series: IFIP International Federation for Information Processing*, volume 262, pages 51–59. Boston:Springer, June 2008.

- [188] Emin Islam Tath, Dirk Stegemann, and Stefan Lucks. Dynamic Anonymity. In *Proceedings of the 4th World Enformatika Conference: International Conference on Information Security, WEC'05*, June 2005.
- [189] Emin Islam Tath, Dirk Stegemann, and Stefan Lucks. Security Challenges of Location-Aware Mobile Business. In *Proceedings of the 2nd International Workshop on Mobile Commerce and Services*, München, Germany, 19 July 2005. IEEE Computer Society.
- [190] Emin Islam Tath, Dirk Stegemann, and Stefan Lucks. Dynamic Anonymity with Mixing. In *Technical Report, University of Mannheim*, March 2006.
- [191] Torsten J. Gerpott. Wettbewerbsstrategische Positionierung von Mobilfunknetzbetreibern im Mobile Business(in German). In: *Silberer, Günter / Wohlfahrt, Jens/ Wilhelm, Torsten (Hrsg.), Mobile Commerce. Grundlagen, Geschäftsmodelle, Erfolgsfaktoren*, pages 43–63, 2002.
- [192] Jo Twist. Blogger grounded by her airline. *BBC News*, 27 October 2004. <http://news.bbc.co.uk/2/hi/technology/3955913.stm>.
- [193] Florent Jacquemard und Daniel Le Mtayer. Rapport Technique EVA No 1 & 2, Langage de speciation de protocoles e cryptographiques de EVA. <http://www-eva.imag.fr/fournitures1.html>, November 2001.
- [194] Ramaprasad Unni and Robert Harmon. Perceived effectiveness of push vs. pull mobile location-based advertising. *Journal of Interactive Advertising*, 7(2), 2007.
- [195] J. Viega, J. T. Bloch, and P. Chandra. Applying Aspect-Oriented Programming to Security. *Cutter IT Journal*, 2001.
- [196] Roy Want, Andy Hopper, Veronica Falco, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.
- [197] Rüdiger Weis and Stefan Lucks. Standardmässige Wave-LAN Unsicherheit. *Datenschutz und Datensicherheit*, 25(11), 2001.
- [198] Bart De Win. *Engineering application-level security through aspect-oriented software development*. PhD in computer science, Department of Computer Science, Catholic University of Leuven, 2004.

- [199] Bart De Win, Wouter Joosen, and Frank Piessens. Developing secure applications through aspect-oriented programming. In *Aspect-Oriented Software Development*, pages 633–650. Addison-Wesley, 2005.
- [200] Bart De Win, Frank Piessens, Wouter Joosen, and Tine Verhanneman. On the importance of the separation-of-concerns principle in secure software engineering. In *In Proceedings of the ACSA Workshop on the Application of Engineering Principles to System Security Design*, 2003.
- [201] Bart De Win, Bart Vanhaute, and Bart De Decker. Security Through Aspect-Oriented Programming. In *Proceedings of the IFIP TC11 WG11.4 First Annual Working Conference on Network Security*, pages 125–138, Deventer, The Netherlands, 2001. Kluwer, B.V.
- [202] C. Wullems, O. Pozzobon, and K. Kubik. Trust your Receiver? Enhancing Location Security. 2004. <http://www.gpsworld.com/gpsworld/article/articleDetail.jsp?id=128320>.
- [203] Michael Juntao Yuan. *Enterprise J2ME: Developing Mobile Java Applications*. Pearson Education, Inc., 2004.
- [204] Erik Zenner, Rüdiger Weis, and Stefan Lucks. Sicherheit des GSM-Verschlüsselungsstandards A5. *Datenschutz und Datensicherheit*, 24(7), 2000.

\* All online citations accessed November 25, 2008.





# Appendix A

## Acronyms

<b>AEAD</b>	Authenticated Encryption with Associated Data
<b>API</b>	Application Programming Interface
<b>APPEL</b>	A P3P Preference Exchange Language
<b>ARM</b>	Advanced RISC Machine
<b>ASCII</b>	American Standard Code for Information Interchange
<b>B2B</b>	Business to Business
<b>B2C</b>	Business to Client
<b>CDC</b>	Connected Device Configuration
<b>CIA</b>	Confidentiality Integrity Availability
<b>CO</b>	Cryptographic Object
<b>CPU</b>	Central Processing Unit
<b>DoS</b>	Denial of Service
<b>EBNF</b>	Extended Backus-Naur-Form
<b>EPAL</b>	The Enterprise Privacy Authorization Language
<b>E-BUSINESS</b>	Electronic Business
<b>E-COMMERCE</b>	Electronic Commerce

---

<b>E-P3P</b>	Enterprise Privacy Practices
<b>FIDIS</b>	Future of IDentity in the Information Society
<b>GEM</b>	Generic Environment for Mobile Business
<b>GPRS</b>	General Packet Radio Service
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ISDN</b>	Integrated Services Digital Network
<b>I/O</b>	Input/Output
<b>ISO</b>	International Organization for Standardization
<b>LACODA</b>	Language for Code Generation and Protocol Analysis
<b>LAMBADA</b>	Location-Aware Mobile Business Adhoc Architecture
<b>TLS</b>	Transport Layer Security
<b>MAC</b>	Message Authentication Code
<b>M-BUSINESS</b>	Mobile Business
<b>MCM</b>	Mix-net Client Manager
<b>M-COMMERCE</b>	Mobile Commerce
<b>PC</b>	Personal Computer
<b>PDA</b>	Personal Digital Assistant
<b>PET</b>	Privacy Enhancing Technologies
<b>PIN</b>	Personal Identification Number
<b>PRIME</b>	Privacy and Identity Management for Europe
<b>P3P</b>	Platform for Privacy Preferences
<b>RPC</b>	Remote Procedure Call

---

<b>SALSA</b>	Software Architectures For Location-Specific Transactions in Mobile Commerce
<b>SPKI</b>	Simple Public Key Infrastructure
<b>SPORE</b>	Security Protocols Open Repository
<b>SSL</b>	Secure Socket Layer
<b>SSO</b>	Single Sign On
<b>TAN</b>	Trans-Aktions-Nummer
<b>TLS</b>	Transport Layer Security
<b>UAProf</b>	User Agent Profile
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>WEP</b>	Wired Equivalent Privacy
<b>WLAN</b>	Wireless Local Area Network
<b>WPA</b>	Wi-Fi Protected Access
<b>WS-SECURITY</b>	Web Services Security
<b>W3C</b>	The World Wide Web Consortium
<b>XML</b>	Extensible Markup Language
<b>XSS</b>	Cross Site Scripting



## Appendix B

# LaCoDa Sample Output Code

### B.1 Class File for EncryptThenAuthenticate\_A.java

```
import cryptort.*;
import cryptort.exceptions.*;
public class EncryptThenAuthenticate_A extends cryptort.CryptoSystem {
    public EncryptThenAuthenticate_A() {
        super();
    }
    public void init() { }
    private OMAC KAut;
    public void setKAut(OMAC ref) {
        this.KAut = ref;
    }
    private Party B;
    public void setB(Party ref) {
        this.B = ref;
    }
    private BitStream send_message;
    public void setsend_message(BitStream ref) {
        this.send_message = ref;
    }
    private AES kEnc;
    public void setkEnc(AES ref) {
        this.kEnc = ref;
    }
}
```

```

    }
    public void doProtocol()
        throws java.beans.PropertyVetoException,
        EnsureFailedException, UnmarshallingException,
        ParseException {
        BitStream ciphertext;
        this.setState(1);
        ciphertext = kEnc.encrypt(send_message);
        this.setState(2);
        this.getB().send(ciphertext);
        this.setState(3);
        this.getB().send(KAut.MAC_generate(ciphertext));
        this.setState(4);
        // nothing to do in this state for this Party
        this.setState(5);
        // nothing to do in this state for this Party
    }
}

```

## B.2 Class File for EncryptThenAuthenticate\_B.java

```

import cryptort.*;
import cryptort.exceptions.*;
public class EncryptThenAuthenticate_B extends cryptort.CryptoSystem {
    public EncryptThenAuthenticate_B() {
        super();
    }
    public void init() { }
    private OMAC KAut;
    public void setKAut(OMAC ref) {
        this.KAut = ref;
    }
    private BitStream receive_message;
    public BitStream getreceive_message() {
        return this.receive_message;
    }
    private Party A;
    public void setA(Party ref) {
        this.A = ref;
    }
}

```

```
    }
    private AES kEnc;
    public void setkEnc(AES ref) {
        this.kEnc = ref;
    }
    public void doProtocol()
        throws java.beans.PropertyVetoException,
        EnsureFailedException, UnmarshallingException,
        ParseException {
        BitStream c_txt;
        M.Tag aut_tag;
        this.setState(1);
        // nothing to do in this state for this Party
        this.setState(2);
        BitStream bs_2 = this.getA().receive();
        c_txt = bs_2;
        this.setState(3);
        BitStream bs_3 = this.getA().receive();
        aut_tag = M.Tag.getFromBitStream(bs_3);
        if (!(aut_tag.getSize() >= 40))
            throw new EnsureFailedException(1, "aut_tag");
        if (!bs_3.isEmpty())
            throw new UnmarshallingException
                ("possible buffer overflow attack");
        this.setState(4);
        if (KAut.MAC_verify(c_txt, aut_tag) == 0) {
            throw new ParseException(1);
        }
        this.setState(5);
        receive_message = kEnc.decrypt(c_txt);
    }
}
```





## Appendix C

# Extended Backus-Naur-Form

The formal notation of the LaCoDa specification language is given in EBNF form in this appendix. The Table C.1 lists the reserved words of the LaCoDa language. In addition, the Table C.2 explains the symbols and operators used in the EBNF notation.

In the EBNF notation, the reserved words are written in typewriter font (Example: “RESERVEDWORD”) and the symbols and operators are written in bold font (Example: “**OPERATOR**”).

PARAM	VAR	FUNCTION
RETURN	SYSTEM	OPTIONS
COMPILER	PARSER	DEF
IN	OUT	INOUT
BLOCK	BEGIN	END
IF	THEN	ELSE
ENDIF	ASSERT	ENSURE
FAIL	NEW	TEMPLATE
NOT	SHARED	MESSAGES
REPEAT	UNTIL	

Table C.1: The Reserved Words

	Either-Or relation
?	optional element
+	one or more instances exist
*	none, one or more instances exist
ASSIGN	:=
ARROW	->
SEMI	;
COLON	:
COMMA	,
DOT	.
LAREA	{
RAREA	}
LRND	(
RRND	)
LSQ	[
RSQ	]
GT	>
LT	<
EQ	=
LE	<=
GE	>=
NE	<>
PLUS	+
MINUS	-
MUL	*
DIV	/
MOD	%
POWER	^
AND	&&
OR	

Table C.2: The symbols and operators used in the EBNF specification



```

45 ParameterBlock := PARAM VariableWithoutMethodInvocation LAREA
46   ( ParameterDeclaration | SharedDeclaration ) *
47   RAREA
48 ParameterDeclaration := ( IN | OUT | INOUT )
49   VariableWithoutMethodInvocation
50   ( COMMA VariableWithoutMethodInvocation ) * COLON
51   IdentifierWithDots SEMI
52 SharedDeclaration := SHARED VariableWithoutMethodInvocation
53   ( COMMA VariableWithoutMethodInvocation ) * SEMI
54 VariableBlock := VAR ( VariableWithoutMethodInvocation ) LAREA
55   ( VariableDeclaration ) *
56   RAREA
57 VariableDeclaration := VariableWithoutMethodInvocation
58   ( COMMA VariableWithoutMethodInvocation ) * COLON
59   IdentifierWithDots SEMI
60 MessageOrRepeat := Message | Repeat
61 Message := ExternalMessage | InternalMessage
62 RepeatPart := Identifier
63 RepeatCondition := UNTIL Identifier COLON OrExpression
64 Repeat := LRND RepeatParty ( COLON RepeatParty ) * RRND COLON
65   REPEAT
66   ( MessageOrRepeat ) * ( RepeatCondition ) +
67   END REPEAT SEMI
68 InternalMessage := Identifier COLON ExpressionOrIfStatement
69 ExternalMessage := Identifier ARROW Identifier COLON
70   Expression ARROW VariableNoObject SEMI
71 VariableNoObject := Identifier
72 ExpressionOrIfStatement := FailStatement | IfStatement
73   | ( Expression SEMI )
74 ExpressionOrIfStatementForIf := FailStatement | IfStatement
75   | ( Expression SEMI )
76 FailStatement := FAIL LRND Expression RRND SEMI
77 BlockStatement := SymbolTable BLOCK BlockVar
78   BEGIN Begin
79   END BLOCK
80 Begin := ( Message ) *
81 BlockVar := ( BlockVariableDeclaration ) *
82 BlockVariableDeclaration := VariableWithoutMethodInvocation
83   COLON VariableWithoutMethodInvocation COLON
84   IdentifierWithDots SEMI
85 IfStatement := IF IfCondition THEN
86   ThenBlock ( ELSE ElseBlock ) ?
87   ENDIF
88 IfCondition := OrExpression
89

```

---

```

90 ThenBlock := ( ExpressionOrIfStatementForIf )
91   (SEMI ExpressionOrIfStatementForIf )* ( SEMI )?
92 ElseBlock := ( ExpressionOrIfStatement )
93   ( SEMI ExpressionOrIfStatementForIf )* ( SEMI )?
94 Expression := ( AssignExpression | OrExpression )
95 AssignExpression := VariableWithoutMethodInvocationOrList
96   ASSIGN InstantiationExpression
97 InstantiationExpression := ( NEW Identifier IRND ArgumentList RRND )
98   | ( OrExpression )
99 OrExpression := AndExpression ( OR AndExpression )*
100 AndExpression := EqualExpression ( AND EqualExpression )*
101 EqualExpression := RelationalExpression
102   ( ( NE RelationalExpression ) | ( EQ RelationalExpression ) ) *
103 RelationalExpression := AdditiveExpression
104   ( ( LT AdditiveExpression ) | ( GT AdditiveExpression ) |
105     ( LE AdditiveExpression ) | ( GE AdditiveExpression ) ) *
106 AdditiveExpression := MultiplicativeExpression
107   ( ( PLUS MultiplicativeExpression ) |
108     ( MINUS MultiplicativeExpression ) ) *
109 MultiplicativeExpression := UnaryExpression
110   ( ( MUL UnaryExpression ) | ( DIV UnaryExpression ) |
111     ( MOD UnaryExpression ) ) *
112 UnaryExpression := ( PLUS UnaryExpression ) |
113   ( MINUS UnaryExpression ) | ( NOT UnaryExpression ) |
114   UnaryExpressionNotPlusMinus
115 UnaryExpressionNotPlusMinus := ListExpression | AnyConstant |
116   VariableWithMethodInvocation | IRND InstantiationExpression RRND
117 ArgumentList := ( Argument ( COMMA Argument )* )?
118 Argument := InstantiationExpression
119 ListExpression := LSQ VariableWithoutMethodInvocation
120   ( COMMA VariableWithoutMethodInvocation )* RSQ
121 VariableWithoutMethodInvocationOrList := ListExpression |
122   VariableWithoutMethodInvocation
123 VariableWithoutMethodInvocation := Identifier
124 VariableWithMethodInvocation := Identifier ( ( MethodInvocation ) )?
125 MethodInvocation := DOT Identifier IRND ArgumentList RRND
126   ( MethodInvocation )?
127 AnyConstant := INTEGER_LITERAL
128 StringOrIdentifier := StringLiteral | Identifier
129 StringLiteral := STRING_LITERAL
130 Identifier := IDENTIFIER1
131 IdentifierWithDots := IDENTIFIER1 ( DOT IDENTIFIER1 ) *
132 IdentifierWithDotsOrConstant := INTEGER_LITERAL | IdentifierWithDots
133 RelationSymbol := EQ | LT | GT | LE | GE | NE

```

---